Extraction of Hepatic Veins in Contrast Enhanced CT with Application to Interventional Planning

Vom Fachbereich Informatik
der Technischen Universität Darmstadt
genehmigte

DISSERTATION

zur Erlangung des akademischen Grades eines
Doktor-Ingenieurs (Dr.-Ing.)
von

Dipl.-Inform. Klaus Drechsler

geboren in Frankfurt am Main, Deutschland

Referenten der Arbeit:	Prof. Dr.-Ing. Georgios Sakas Technische Universität Darmstadt Prof. Dr. techn. Dieter W. Fellner Technische Universität Darmstadt Priv.-Doz. Dr. med. Dr. med. habil. Christian Mönch Johann Wolfgang Goethe-Universität
Tag der Einreichung:	23/08/2012
Tag der mündlichen Prüfung:	10/10/2012

Darmstädter Dissertation
D 17
Darmstadt, 2012

Erklärung zur Dissertation

Hiermit versichere ich die vorliegende Dissertation selbständig nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den 23. 08. 2012 Klaus Drechsler

Acknowledgement

I would like to thank a number of people who helped me through the completion of this thesis. Foremost, I would like to express my sincere gratitude to my doctoral advisor and mentor, Prof. Georgios Sakas, for his continuous support of my research, for his straightforwardness and honesty, his motivation and inspiration. His guidance greatly helped to keep my thesis on track.

Aside from my doctoral advisor, I would like to sincerely thank the two further referees, Prof. Dieter W. Fellner and Dr. Christian Mönch for their willingness to review my work and their insightful comments.

My thanks also go to Dr. Arjan Kuiper for proofreading my thesis and Dr. Stefan Wesarg for his valuable last minute comments regarding presentation of my work.

In addition, I would like to thank all my fellow workers from the Department of Cognitive Computing & Medical Imaging at Fraunhofer IGD in Darmstadt for the friendly atmosphere and fruitful discussions about all kind of topics. Especially, I want to thank Cristina Oyarzun Laura, Martin Hoßbach and Marius Erdt for our leisure time activities, which helped most of the time to keep work and life in balance.

Furthermore, I am obliged to my students for helping me to implement several methods which contributed to the success of my thesis. Primarily, I would like to express my gratitude to Simon Oelmann for his long-term support and critical discussions. Furthermore, I would like to thank Christopher Kreitz, Michael Strosche and Steven Meixner for their contributions.

A heartfelt thanks go to my beloved girlfriend Cristina for her understanding when I spent more time with LaTeX than with her and for her continuous encouragement to finish my thesis.

Last but not least, I would like to express my deepest gratitude to my parents for their love, dedication and support. Obviously, without you, this thesis would have never happened. Thank you so much!

Abstract

The Liver performs several important tasks that are essential for survival. However, liver cancer, the third most common type of cancer, affects these functions significantly. Different treatment options are available, but a surgical resection, if possible, offers the best prognosis for the patient. Thus, the decision, whether a surgical resection is feasible, is important and must be taken with care in a pre-interventional planning stage.

Modern volumetric imaging techniques such as CT or magnetic resonance imaging (MRI) are utilized to decide which treatment is best for the patient and to plan the intervention. However, the amount of anatomical details visible in the acquired volumes is steadily increasing. This comes along with an increasing amount of data per patient. Manual examination is time consuming and prone to errors. As a matter of fact, several software systems were proposed to support the surgeon during the planning phase.

The extraction of blood vessels plays an important role in these applications. The segmentation of vessels is a challenging problem that has to deal with acquisition-dependent problems such as noise, contrast, spatial resolution, and artifacts. Furthermore, blood vessel specific characteristics like high variability of size and curvature result in additional difficulties for segmentation algorithms. The liver, in particular, exhibits another challenge to vessel segmentation algorithms. Its supply and drain vessel systems are densely distributed within the liver, and because of partial volume effects and motion artifacts, they seem to be connected at some points.

The focus of the present thesis is the robust extraction of hepatic veins in multiphase CT volumes. Therefore, an image processing pipeline is presented that covers vessel enhancement, vessel segmentation, graph creation and tree reconstruction. The pipeline was used to develop an application for interventional planning. It allows for the simulation of intraoperative hepatic vein clamping for (sub-)segment oriented liver resections and the execution of risk analysis to judge surgical risk during an atypical resection. Furthermore, results of the present thesis were also successfully used in an application for intraoperative navigation to extract liver vessels in 3D ultrasound data and matching of anatomical vessel trees and graphs of the liver for registration of 3D volumes.

Zusammenfassung

Krebs ist die zweithäufigste Todesursache weltweit, welche 2008 für 7.6 Millionen Todesfälle verantwortlich war. Davon traten 70% in Ländern der dritten Welt auf. Laut Angaben der World Health Organization (WHO) ist Leberkrebs die dritthäufigste Krebsart und jährlich für 700.000 Todesfälle verantwortlich. Das Leberzellkarzinom (Hepatozelluläres Karzinom) ist die häufigste primäre Leberkrebsart. Dieser Krebstyp entwickelt sich nach einer viralen Hepatitis B/C Infektion oder einer Leberzirrhose. Die Leberzirrhose entwickelt sich als Folge einer chronischen Leberkrankheit die beispielsweise durch Alkoholmissbrauch verursacht wird. Charakteristisch für eine Leberzirrhose ist, daß gesundes Lebergewebe durch narbiges Gewebe und Knötchen ersetzt wird und dadurch die Leberfunktion einschränkt. Ungefähr 50% aller Todesfälle, die durch das Leberzellkarzinom verursacht werden, treten in China auf. In westlichen Ländern sind Metastasen von Tumoren anderer Organe die häufigste Ursache von Leberkrebs. In Deutschland wurden im Jahr 2010 219.000 Todesfälle aufgrund von Krebs registriert. Nur ca. 6000 Menschen entwickeln jährlich ein Leberzellkarzinom, allerdings mit steigender Tendenz.

Die Leber ist für das Überleben äußerst wichtig. Sie spielt eine wichtige Rolle beim Stoffwechsel und ist außerdem an der Proteinsynthese, Glycogenspeicherung, Hormonproduktion und der Produktion von Verdauugssäften beteiligt. Im häufig verwendeten Couinaud-System kann die Leber in acht Segmente eingeteilt werden, die unabhängig voneinander von der Leberarterie, den Lebervenen und Gallengängen durchzogen sind. Es existiert derzeit keine Möglichkeit, um das Fehlen dieser wichtigen Funktionen zu kompensieren. Aus diesem Grund ist es wichtig, eine Lebererkrankung frühzeitig und richtig zu diagnostizieren und die gewonnenen Informationen zu verwenden, um die bestmögliche Behandlungsstrategie zu wählen.

Die derzeit am häufigsten eingesetzte Methode zur Diagnose eines Leberzellkarzinom ist ein mehrphasiger Computertomographie-Scan des Abdomens [GKL*09]. Dabei wird üblicherweise ein Kontrastmittel in eine Armvene intravenös injiziert. Das Kontrastmittel erreicht dann die Leber zu zwei unterschiedlichen Zeitpunkten. Das erste Mal durch die Leberarterie (Arterielle Phase) und beim zweiten Mal durch die Portalvene (Portalvenöse Phase). Während dieser Phasen ändert der Tumor sein Aussehen und zeigt einige spezifischen Eigenschaften, die auf die Krebsart schliessen lassen.

Die zur Verfügung stehenden Behandlungsmäglichkeiten richten sich nach Art und Größe des Tumors. Eine chirurgische Resektion bietet dabei die beste Prognose für den Patienten, von denen aber nur ca. 10-15% in Frage kommen. Bei einer chirurgischen Leberresektion unterscheidet man zwischen typischen Resektionen entlang Segmentgrenzen und atypischen Resektionen bei denen die Schnittgrenzen tumororientiert verlaufen und beliebige Formen annehmen können. Typische Resektionen werden normalerweise bevorzugt, da hierbei das Risiko für Blutungen geringer ist. Aus medizinischer Sicht kann aber auch in ausgewählten Fällen eine atypische Resektion sinnvoll sein. Es

ist überaus wichtig, vor der Behandlung eines Patienten zu evaluieren, ob eine Resektion möglich ist. Dabei helfen moderne bildgebende Verfahren, wie die Computertomographie und Magnetresonanztomographie. Sie ermöglichen die räumlichen Lagebeziehungen wichtiger Strukturen einzuschätzen, sowie die Ausbreitung und Art des Tumors zu bestimmen.

Stetige Verbesserungen der räumlichen und zeitlichen Auflösung ermöglichen immer mehr Details darzustellen. Das führt allerdings auch zu einer erhöhten Datenmenge pro Patient, die vom Radiologen verarbeitet werden muß. Um Fehler zu vermeiden und um den Chirurgen bei der Diagnose zu unterstützen werden technische Hilfsmittel benötigt, die einen Teil der anfallenden Daten automatisiert verarbeiten können.

Abhängig vom medizinischen Anwendungsfall sind unterschiedliche anatomische Strukturen von Interesse. Im Besonderen spielt das Extrahieren von Blutgefäßen eine wichtige Rolle. Beispielsweise lassen sich dadurch Stenosen der Koronararterien detektieren. Im Fall der Leber werden Gefäße für die Planung von chirurgischen Resektionen benötigt. Sie ermöglichen das Berechnen der Couinaud Segmente, sowie eine automatisierte Analyse des Risikos bei einer atypischen Resektion. Außerdem werden Gefäße als Landmarken für die Registrierung zweier Volumen verwendet. Das ermöglicht zusätzliche Informationen aus unterschiedlichen Modalitäten zu einem Gesamtbild zu fusionieren. Aber auch die Nutzung für die intraoperative Navigation ist derzeit Gegenstand vieler Forschungsarbeiten.

Die manuelle Segmentierung von Blutgefäßen ist langwierig und fehleranfällig. Automatisierte Segmentierungsverfahren sind deshalb wichtige Hilfsmittel für die klinische Praxis. Allerdings erschweren aufnahmeabhängige Probleme wie Rauschen, schwankende Kontraste, unterschiedliche Auflösungen und Artefakte den Einsatz solcher Verfahren. Dazu kommen spezifische Probleme, wie beispielsweise variable Gefäßdurchmesser, die die den Routineeinsatz noch zusätzlich erschweren. Die Entwicklung robuster Verfahren ist daher Gegenstand vieler Forschungsarbeiten und ist nach wie vor nicht zufriedenstellend gelöst. Darüber hinaus stellt die Leber Gefäßsegmentierungsverfahren vor eine zusätzliche Herausforderung. Aufgrund von Partialvolumeneffekten und Bewegungsartefakten scheinen Portalvene und Lebervene an einigen Stellen miteinander verbunden zu sein. Es ist wünschenswert und für einige Anwendungen sogar erforderlich, diese beiden Gefäßsystem voneinander zu trennen. Das ermøglicht dann die getrennte Visualisierung dieser beiden Gefäßsysteme, sowie die automatisierte Weiterverarbeitung. Einige Forschungsarbeiten wurden zu diesem Thema bereits veröffentlicht. Jedoch ist auch dieses Problem noch nicht zufriedenstellend gelöst.

Der Fokus dieser Arbeit ist die robuste Extraktion von Lebervenen aus kontrastverstärkten CT Volumen. Die resultierende Verfahren wurde verwendet, um eine Applikation zur Planung chirurgischer Leberresektionen zu entwickeln. Darüber hinaus werden Teile des entwickelten Verfahrens in einem System zur intraoperativen Navigation in der Leberchirurgie eingesetzt, um Gefäße in 3D Ultraschalldaten zu extrahieren [OLDE*12, KLDW12] und korrespondierende Bifurkationen in Gefäßgraphen zu detektieren [OLD11a, OLD11b, OLD11c].

Die wesentlichen Beiträge dieser Arbeit sind:

Gefäßextraktion

1. Ein Klassifikationsschema für Gefäßextraktionsmethoden wird vorgeschlagen, welches Verfahren auch anhand der Validierungsmethodik klassififiziert.
2. Ein Vergleich von drei konzeptionell unterschiedlichen *vesselness* Funktionen für die Hervorhebung von Gefäßstrukturen.
3. Ein multiskalen Ansatz zur Integration von Teilergebnissen, welcher die Representation von Gefäßen im Skalenraum berücksichtigt.
4. Ein neuer Filter zur Hervorhebung von Gefäßstrukturen.
5. Ein regionenbasiertes Segmentierungsverfahren, welches Wellenfronten verwendet, um Ausläufe in benachbarte Regionen zu verhindern.
6. Ein Filter, um Löcher in Segmentierungsergebnissen zu füllen, ohne die Form der Gefäße signifikant zu beeinträchtigen.
7. Ein dreistufiges Verfahren, um Gefäßskelette in Graphen zu überführen.
8. Ein Verfahren, um Lebervenen voneinander zu trennen.

Evaluationsmethodik

1. Ein Framework zum standartisierten Evaluieren von Gefäßsegmentierungsalgorithmen

Anwendung

1. Eine Anwendung zur interventionellen Planung von typischen und atypischen Leberresektionen.
2. Ein Graph-cut Verfahren zur Tumorsegmentierung.

Contents

List of Figures

List of Tables

1. Introduction

1.1. Motivation

Cancer is the second leading cause of mortality worldwide. It led to 7.6 million deaths (approximately 13%) in 2008, of which 70% occurred in low- and middle-income countries. According to the World Health Organization (WHO), liver cancer is the third most common type of cancer (Figure 1.1), causing 700,000 deaths annually. Hepatocellular carcinoma (HCC) is the most common type of primary liver cancer. It usually develops after either a viral Hepatitis B/C infection (20%) or a liver cirrhosis (80%). Cirrhosis is a consequence of a chronic liver disease caused, for example, by alcohol abuse. It is characterized by replacement of healthy liver tissue with fibrosis, scar tissue, and nodules. About 50% of the deaths caused by HCC occurred in China. In Western countries, the most frequent liver tumors are metastasis. In Germany, 219,000 deaths (25.5%) were caused by cancer in 2010. Only 6000 people develop HCC per annum, but this is increasing.

The liver is essential for survival. It plays a major role in metabolism and is involved in detoxification, protein synthesis, glycogen storage, hormone production, and production of biochemicals necessary for digestion. In the widely used Couinaud system, the liver can be divided into eight functional lobes, which are independently supplied by its artery, veins, and bile duct. There is currently no long-term method of compensating for the absence of these functions. For this reason, it is important to diagnose the disease correctly at an early stage and choose the best treatment option available.

A multiphase computed tomography (CT) scan of the abdomen is currently the best method to diagnose HCC [GKL*09]. Therefore, a contrast agent (COA) is injected intravenously (usually in an arm vein), reaching the liver twice. The first time is through the liver artery (arterial phase) and the second time is through the portal vein (venous phase). During these phases, the tumor changes its appearance and shows some key characteristics that reveal its true nature.

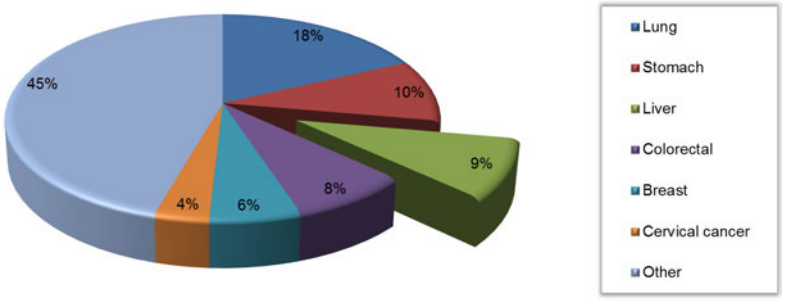


Figure 1.1.: Cancer deaths 2008. Liver cancer is the third most common type.

Treatment options depend on tumor size and staging (e.g., size, spread, involvement of vessels, metastasis, etc.). A (partial) surgical resection (hepatectomy) offers the best prognosis for long-term survival, but this kind of treatment is suitable for only 10-15% of patients. Some other options are radiofrequency ablation (RFA), cryosurgery, high intensity focused ultrasound (HIFU), and transcatheter arterial chemoembolization (TACE). If a hepatectomy is feasible for a particular case, it may be anatomic, i.e., one or more functional segments are completely resected, or non-anatomic, i.e., arbitrary pieces are resected. The former is usually preferred due to the lower risk of bleeding. However, in selected cases, the latter can also be safely performed. The decision, whether a surgical resection is feasible is crucial and is a decision between life and death in the long term. It must be taken with care in a pre-interventional planning stage.

1.2. Problem statement

Modern volumetric imaging techniques such as CT or magnetic resonance imaging (MRI), aid in the understand of a patient's anatomy and pathologies. For example, by looking at the acquired data, physicians determine the degree of tumor spread, the type of tumor, where it is located, and which major structures are close to the tumor. This information can be used to decide which treatment is best for the patient and to plan the intervention. The amount of anatomical details visible in the acquired volumes is steadily increasing. However, high spatial and temporal resolutions and broad use of the mentioned imaging techniques lead to an increasing amount of data per patient. As a matter of fact, more data has to be examined by a physician. If done manually, this is time consuming and prone to errors. As a consequence, tools are needed that process the vast amount of data automatically, to a certain degree, and support the physician in the diagnosis of diseases and planning of required treatment.

Depending on the medical use case, various anatomical structures are of interest, and each one introduces new challenges for segmentation algorithms. In particular, the extraction of blood vessels plays an important role in several applications. In the case of the human heart, some examples are the automated detection of stenoses in coronary arteries [WF04, WKF06, WKJL08] and the vessel-based 2D-3D registration of preoperatively acquired computed tomography angiograms with coronary angiograms of the heart [LC06, TLG*05, DOL10b] for support during intervention [DOLS09, DOL11]. In the case of the human liver, vessels are used to approximate liver segments [SPSP02, BPJ*04], plan surgical interventions [SPSP02, RBBS06, DZF*10, SSS*10], perform risk analysis [HZH*09, SMS*10], and carry out 3D-3D registrations between different modalities where vessels serve as landmarks for intraoperative navigation [OLD11a, DOLCE10, NKLR10, LPH*09, ZAGW07].

The aforementioned applications have in common that they require the segmentation of blood vessels as a preprocessing step. However, manual delineation of blood vessels in volumetric images is error prone and time consuming. In cases where high processing speed is a requirement, such as in intervention support, it is not feasible at all. Automated vessel segmentation algorithms add value to the clinical workflow by taking over this tedious work.

Vessel segmentation is a challenging problem that has to deal with acquisition-dependent problems such as noise, contrast, spatial resolution, and artifacts. Furthermore, blood vessel specific characteristics like high variability of size and curvature result in additional difficulties for segmentation algorithms. A great deal of research is dedicated to vessel segmentation. However, the problem has yet to be solved in a satisfactory way.

The liver, in particular, exhibits another challenge to vessel segmentation algorithms. Its supply and drain vessel systems are densely distributed within the liver, and because of partial volume effects and motion artifacts, they seem to be connected at some points. It is desirable to separate those vessel systems during or after segmentation. This allows for a delineated visualization of those vessel systems and enables some of the aforementioned applications, including automated risk analysis and segment approximation. Some research was carried out to automate this task. However, the separation of interconnected vessel systems is still challenging. One reason is that segmented vessels do not necessarily follow a clear model assumption. For instance, it is not guaranteed that segmented vessels have a decreasing diameter if traversal starts at the root.

1.3. Contributions

The focus of the present thesis is the robust extraction of hepatic veins in multiphase CT volumes. The results were used to develop an application for interventional planning. Such planning allows for the simulation of intraoperative hepatic vein clamping for (sub-)segment oriented liver resection and the execution of risk analysis to judge surgical risk during an atypical resection. Results of the present thesis were also successfully used in an application for intraoperative navigation to extract liver vessels in 3D ultrasound data [OLDE*12, KLDW12] and matching of anatomical vessel trees and graphs of the liver [OLD11a, OLD11b, OLD11c].

The main contributions of this thesis are summarized as follows:

Vessel extraction

1. An algorithmic classification scheme for vessel extraction methods is proposed. Its novelty is that it also classifies methods according to their validation methodology.

2. A comparison of three conceptual different vesselness functions for multiscale analysis of the liver vasculature was carried out and recommendations for their applications are given.

3. A multiscale integration approach is proposed that makes use of the scale space representation of vessels in order to enhance vessels toward their centers. Using this approach, nearby vessels are better delineated and segmentation becomes significantly more robust to noise.

4. A new vessel enhancement filter for tubular structures based on the Laplacian is proposed to enhance detect vessels in medical images. The Laplacian can be directly extracted from the Hessian matrix without any additional calculations. It is

proven that it produces results equivalent to other filters. Furthermore, it is used to reduce 50% of the eigenvalue calculations.

5. A wavefront propagation technique is proposed that prevents leaks by imposing a threshold on the maximum number of voxels that the propagating front must have.

6. A cascaded voting filter is proposed to fill holes, bays, and tunnels without overly affecting the shape of the vessels. These defects are caused at locations where the tubular structure assumption is violated.

7. A three-stage process to transform a skeleton into a graph is proposed. It consists of a new skeleton classification scheme, skeleton traversal using a deterministic finite automaton and the decomposition of vessel skeletons into sub-branches for efficient further processing. Its distinctive feature is that it does not require a preselected root voxel to produce correctly connected sub-branches; it is independent of the traversal direction.

8. An algorithm for the separation of portal- and hepatic veins is presented. It takes into account that segmented vessels do not follow a clear model assumption.

Evaluation methodology

1. A framework for standardized evaluation of vessel segmentation algorithms based on physiological inspired simulated vessel trees was developed. It was used to quantitatively measure the performance of the developed segmentation algorithm.

Application

1. An application for interventional planning was developed to support the surgeons in their decision making process. It is the first application for interventional planning of liver resections that utilizes deformable registration in order to fuse information contained in several datasets. It supports the surgeon during the preparation of anatomical and atypical resections.

2. A multiresolution method based on graph cuts is presented to automatically identify a region of interest (ROI) in a coarse grid and is used in a fine grid to segment the desired object. The method was applied to the segmentation of liver tumors.

1.4. Organization of the thesis

This thesis is organized as follows.

Chapter 2 introduces an algorithm-oriented taxonomy for the classification of vessel extraction methods. Its novelty lies in the inclusion of a taxonomy for validation methodologies. Each taxon is separately explained and the related work is reviewed accordingly. The proposed taxonomy is used to clearly represent current related work in hepatic vein extraction methods in tabular form. The chapter ends with a discussion and identifies open research topics.

Chapter 3 describes the proposed vessel extraction pipeline. In Section 3.2, it starts with preprocessing of CT data to detect and enhance hepatic veins. Therefore, after an introduction to multiscale vessel enhancement filtering, three conceptually different vesselness functions are compared with regard to their ability to separate nearby vessels and their behaviour at junctions. Afterwards, a multiscale integration approach is presented that significantly improves robustness to noise and improves delineation of nearby vessels. Finally, a new vesselness filter based on the divergence of the gradient vector field is developed. It is inspired by filters that use the scale-space gradient to calculate a medialness measure at multiple scales in order to detect vessels. The idea is to measure how much a point behaves like a source or sink for gradient vectors. In a multiscale framework, the divergence of the gradient vector field equals the Laplacian of Gaussian. It is proven that results of the proposed filter produce equally good results as other vesselness filter, but with significantly less calculations for eigenanalysis.

Vessel enhancement filter are based on the assumption that vessels are similar to tubular structures. This model assumption is violated at vessel furcations, resulting in a low vesselness filter response. A region-based segmentation method is usually used for interactive applications due to its ease of use and computational efficiency. When combined with vessel enhancement filters, this leads to holes and tunnels at furcations. To solve for this, Section 3.4 presents a cascaded voting filter as the next step in the processing pipeline. It reliably eliminates those defects without overly affecting the shape of the vessels.

Further processing of the segmented vessels is done on a higher level of abstraction, namely on a formal graph representation of the segmented vessels. Therefore, a three-stage process is presented in Section 3.5. It consists of a new skeleton classification scheme, a deterministic finite automaton for skeleton traversal and the decomposition of vessel skeletons into sub-branches for efficient further processing. Its distinctive feature is that it does not require a preselected root voxel to produce correctly connected sub-branches of the hepatic veins; it is independent of the traversal direction.

In Section 3.6 a method for computer assisted separation of interconnected hepatic veins is presented. It automatically suggests branches to be removed based on model assumptions and processing of the previously generated graph. Compared to similar methods, it takes into account that segmented vessels to not always follow clear model assumptions.

Sections 3.7 deals with the automated evaluation of vessel segmentation methods. Therefore, a evaluation framework is proposed. It generates ground-truth data of vessel trees based on physiological principles and uses them to create simulated CT datasets, which serve as input to the segmentation algorithm under study. The proposed framework processes the results automatically and calculates several image-based and graph-based metrics.

In Section 3.8 evaluation results of the proposed methods are presented. Finally, Chapter 3 is concluded in Section 3.9 with a short discussion.

Chapter 4 describes a proof-of-concept implementation of an interventional planning system. It provides the aforementioned vessels extraction method, tools for tumor segmentation and enables planning of typical and atypical resections. Furthermore, it utilizes a deformable registration method to fuse complementary information spread in multiple

datasets.

Chapter 5 concludes this thesis and gives pointer for future research directions.

2. State of the art

There exist no single vessel segmentation method that is able to extract vessels from every imaging modality, for every application domain in a fully automatic manner. As a matter of fact, a plethora of algorithms were proposed to serve a specific need. In this chapter, a review of state of the art vessel extraction techniques is given. The review focuses mainly, but not exclusively, on the extraction of the liver vasculature as the application domain. A comprehensive and general review of vessel segmentation methods was published by Kirbas et al. [KQ04]. Another comprehensive review with focus on angiographic images was published by Lesage et al. [LABFL09] and Vijayakumari and Suriyanarayanan [VS12] and Fraz et al. [FRH*12] surveyed vessel segmentation methods with focus on retinal images.

The remainder of this chapter is organized as follows. Section 2.1 provides some common definitions that are used throughout the thesis. Section 2.2 introduces a taxonomy to classify available vessel extraction methods. Each taxon is algorithmically introduced and relevant publications are cited. Section 2.2.5 summarizes major research contributions in the field of hepatic vein extractions using the proposed classification scheme. Section 2.3 discusses the pros and cons of available methods and Section 2.4 finally concludes the discussion by identifying the challenges handled in this thesis.

2.1. Notation and definitions

In this section, some common definitions are given which are used throughout the whole thesis. Definitions with a more narrow scope within this thesis are given in the individual sections.

Definition 2.1. An *image I* is a real valued discrete grid defined over its domain Ω in an n-dimensional space. It is notated as

$$I(x) \in \mathbb{R}, \tag{2.1}$$

$$x \in \Omega_n. \tag{2.2}$$

Definition 2.2. A *segmentation* is a discrete valued image over the same domain Ω_n,

$$S(x) \in \mathbb{N}. \tag{2.3}$$

Definition 2.3. Most of the time, *binary* images are produced. This is a special case of the above formula. Thus, a *binary segmentation* is defined as

$$S(x) \in \{0, 1\}. \tag{2.4}$$

Definition 2.4. The set of all segmented voxels is denoted by

$$S = \{x \in \Omega_n | S(x) = 1\} \tag{2.5}$$

Definition 2.5. A *skeleton* is a one voxel thin line of the lumen, which retains topology, lies geometrically in the middle and preserves connectivity.

2.2. Classification scheme

Kirbas et al. [KQ04] proposed a classification into pattern-recognition-based, model-based, tracking-based, artificial intelligence-based, neural network-based and miscellaneous tube detection-based approaches. Each category was further subdivided and partly overlapping. For example, they had separate subgroups for ridge-based methods and tracking-based methods. However, ridge-based methods and most tracking-based methods extract the centerline. It seems natural to group them into the same category. Furthermore, machine learning-based methods were treated as a separate category, although these algorithms only use machine learning as a tool to classify single voxels, determine thresholds and so on. These methods thus can be split and merged with other categories. As part of their review, Lesage et al. [LABFL09] classified vessel segmentation methods from an algorithmic point of view. This follows the idea that actual implementations usually consist of several processing steps, which are interchangeable. The goal of this section is to establish a classification scheme that answers the following questions about a specific method:

1. How does it achieve the vessel extraction algorithmically?
2. On which input does it operate?
3. Which output is generated?
4. How are results validated?

Figure 2.1 illustrates the proposed classification scheme graphically. It consists of four main blocks: Input, Vessel extraction, Output and Validation. To describe a vessel extraction method algorithmically, it is advantageous to divide it into three stages as Lesage et al. [LABFL09] did. In a preprocessing stage (Section 2.2.1), the data is prepared for the following extraction stage (Section 2.2.2), which does the actual extraction of the vessel lumen or centerline. In a postprocessing stage (Section 2.2.3), generated outputs are improved and/or additional outputs generated.

One aspect, which recent survey papers [KQ04, LABFL09] neglected, is the classification of algorithms according to their validation. For this purpose, different methods, from very simple to very complex, exist and published methods vary considerably in the effort spent to answer the question how good a proposed algorithm is. Validation (Section 2.2.4) should be seen as an integral part of algorithm development and as such a part of a classification scheme.

The following sections describe the preprocessing, extraction and postprocessing steps in detail and review the current state of the art.

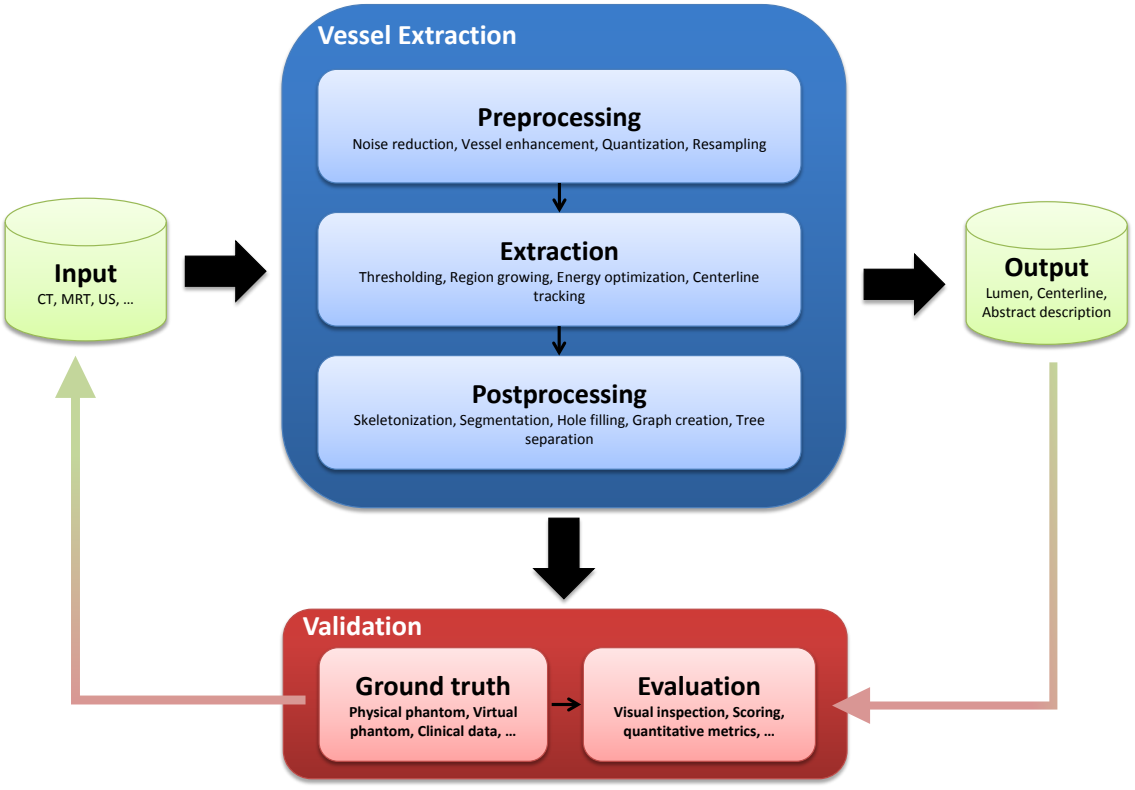


Figure 2.1.: Classification scheme for vessel extraction methods consisting of four building blocks: Input, Vessel extraction, Output and Validation.

2.2.1. Preprocessing

Before vessels are extracted from an image, several preprocessing steps can be carried out to simplify the extraction process or learn image parameters. These preprocessing steps can be further subdivided into the following categories:

- Pre-Segmentation,
- Image resampling and quantization,
- Image analysis
- Noise reduction,
- Vessel enhancement techniques.

2.2.1.1. Pre-Segmentation, image resampling and quantization

Most of the extraction algorithms presented in Section 2.2.2 require a rough region of interest (ROI) to limit the extraction of vessels to a specific region. Usually, an organ mask is first obtained. Examples include the heart [DOL10b], the lung [SMHS09], and the liver [PWL11, PVWL12]. Numerous automatic and semi-automatic methods for liver

segmentation operating in 2D and 3D have been proposed. Amongst them are statistical shape models [HM09], atlas registration [FSK07], level-sets [PD01], graph-cuts [BBB*07], and rule-based systems [SAS*07]. Model-based approaches have been established as one of the most successful methods since they incorporate a-priori information about the expected shape. Therefore, they generally perform well in regions with poor organ boundaries and produce few segmentation artifacts.

A review of general liver segmentation methods can be found in [CC07, CCE09]. A review of state of the art model-based approaches that incorporate statistical shape information can be found in [HM09].

More general preprocessing methods include image resampling to make an anisotropic dataset isotropic [SNA*97, SNS*98] or reduce spatial resolution for faster processing. Image quantization can be used to reduce the amount of data in order to increase processing speed and reduce memory requirements [THMS05].

2.2.1.2. Image analysis

Some of the extraction schemes explained in the next section require some thresholds to function properly. Machine learning methods can be applied to the image as a preprocessing step to obtain these thresholds. Probably, the most popular way to obtain an optimal threshold parameter, is to analyze the gray value distribution of an image using Otsu's method [Ots79], which maximizes the between-class variance. For instance, Beichel et al. [BPJ*04] used Otsu's method to calculate an optimal threshold to separate vessels in a vessel enhanced image from the background.

Other options include machine learning techniques such as statistical analysis. Oliveira et al. [OFC11], for example, calculated a Gaussian mixture model of three Gaussians for liver vessels, nodules and parenchyma. The resulting mixture model was then used to derive thresholds for a region growing algorithm.

2.2.1.3. Noise reduction

All medical image recording devices add undesired noise to the resulting image and degrade the content thereby. This makes further processing challenging. As a matter of fact, preprocessing can include the algorithmic reduction of noise, preferably without changing the image content and without washing out dominant structures like edges. Noise reduction methods can be linear or non-linear. Furthermore, they can be isotropic or anisotropic. A linear isotropic method to reduce noise is gaussian blurring. It is quite effective in homogenous areas, but also washes out edges. This is an undesirable behavior. Thus, noise is often reduced using filtering techniques that are able to preserve edges. Examples include non-linear isotropic methods like median filtering and non-linear anisotropic methods like mean-gaussian filtering and anisotropic diffusion. Noise reduction methods are commonly used as a preprocessing step for vessel extraction. Examples include median filtering [Sel99, SCG*08], mean-gaussian filtering [KTA09] and anisotropic diffusion [HVN08].

Figure 2.2 shows a comparison of median filtering with Gaussian blurring applied to a CT dataset. It can be seen that gaussian filtering blurrs edges, while median filtering

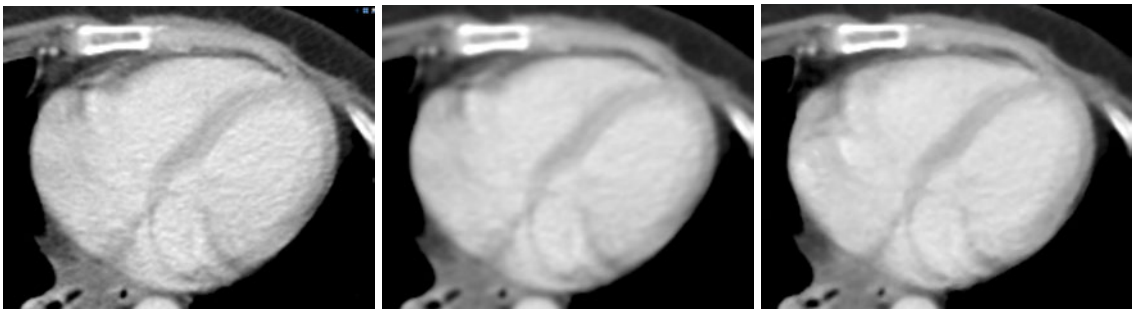


Figure 2.2.: Comparison of Gaussian (middle) with median (right) filtering applied to a CT dataset to reduce noise. The Gaussian kernel was calculated with $\sigma = 1$ and the median kernel had a radius of $r = 1$. Compared to the original image (left), it can be seen that gaussian filtering blurrs edges while median filtering can change the content of the image.

preserves them better. However, median filtering can change image content to a certain degree. Anisotropic schemes are usually computationally demanding, while isotropic schemes may blurr edges or change the content.

2.2.1.4. Vessel enhancement

To improve image quality further, specialized filtering methods are used to detect and enhance vessel-like structures and reduce non-vessel structures. They can be further subdivided into the following groups

- Directional approaches,
- Anisotropic diffusion or
- Morphological methods.

Directional approaches Image features, which provide local orientation of image structures, play an important role. They are often embedded in a multiscale framework in order to detect tubular structures of varying sizes. Directional features can be based on derivative features, like the Hessian or gradient, or on integrative features, like the second-order inertia moments.

Two notions emerged in literature to give derivative-based filters a name: *Vesselness* and *Medialness*. Vesselness filters measure the likeliness of a voxel belonging to a vessel by evaluating features at the current spatial position. Medialness filters measure the likeliness that a voxel is in a medial position of a tubular structure by evaluating features at multiple spatial positions equidistant from the current position. Typically, first-order derivatives (gradient), second-order derivatives (Hessian) or a combination of both are used as features.

Figure 2.3 shows an example to illustrate the idea. The first image is a Maximum Intensity Projection (MIP) of an MRA image. The following four images are filter responses at increasing scales. At each scale, vessels with a radius that correlates with the selected scale are detected and enhanced.

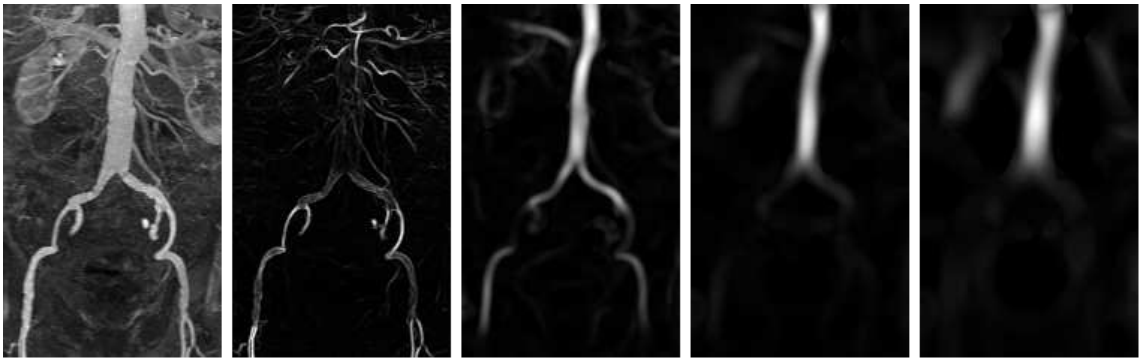


Figure 2.3.: The first image shows a Maximum Intensity Projection of an MRA image. The remaining pictures show vesselness responses with increasing scales. Images taken from [FNVV98].

Eigenvalue	Value	Structure
λ_1, λ_2	< 0	
$\lvert\lambda_1\rvert, \lvert\lambda_2\rvert$	$\gg 0$	bright tube
$\lvert\lambda_3\rvert$	~ 0	
λ_1, λ_2	> 0	
$\lvert\lambda_1\rvert, \lvert\lambda_2\rvert$	$\gg 0$	dark tube
$\lvert\lambda_3\rvert$	~ 0	

Eigenvalue	Value	Structure
λ_1	< 0	
$\lvert\lambda_1\rvert$	$\gg 0$	bright plate
$\lvert\lambda_2\rvert, \lvert\lambda_3\rvert$	~ 0	
λ_1	> 0	
$\lvert\lambda_1\rvert$	$\gg 0$	dark plate
$\lvert\lambda_2\rvert, \lvert\lambda_3\rvert$	~ 0	

Table 2.1.: Eigenvalues of the Hessian matrix can be used to detect tube and plane structures in 3D volumes as described by Lorenz et al. [LCB*97]

Koller et al. [KGSD95] used two shifted and oriented Gaussian derivative kernels to detect parallel vessel walls at multiple scales. They used the eigenvector corresponding to the smallest eigenvalue of the Hessian matrix to orient the filter in 3D. Since then, Hessian-based vessel enhancement filters were further improved and are very popular to date [LCB*97, SNA*97, FNVV98, KMA*00, OBN03, PBB05b, BvAS*07, ZCS*07, ERS08, MHB09, SMHS09, BPS*10, NKM10, YBE11].

Lorenz et al. [LCB*97] investigated the eigenvalues of the Hessian-matrix to detect different structures. They found that eigenvalues can be used to detect tubes and planes. Table 2.1 shows the conditions of the eigenvalues in order to detect these structures. Based on this insight, they constructed a vesselness function to detect tubular structures solely based on the eigenvalues.

Sato et al. [SNA*97, SNS*98] also proposed a vesselness function based on eigenvalues. Their solution is based on an experimental study of filter responses using a tubular model.

Frangi et al. [FNVV98] constructed a vesselness function by geometrically interpreting the eigenvalues as second-order ellipsoid. They introduced two geometric ratios to distinguish blob-, plate-, and line-like structures.

Bennink et al. [BvAS*07] proposed an intensity independent vesselness function which exploits first and second-order derivatives. The main component was a steerable second-order line filter with Canny-like criteria [JU04]. Furthermore, deviations from plate-like structures using the Hessian eigenvalues and the fact that the gradient magnitude equals

0 in the center of a tubular structure were taken into account.

Erdt et al. [ERS08] composed a filter as a linear combination of second order Gaussian basis filters (steerable filter). They analytically determined its coefficients to maximize the output of the convolution of the filter with a tubular model. The solution was used to derive a vesselness function.

Kaftan et al. [KTA09] used the idea of flux maximizing flows to construct a vesselness measure. The idea is based on the observation that the inward flux is maximized when a surface is well aligned with the real vessel. They computed this feature by integrating over the gradient vector field along a spherical surface at multiple scales.

Krissian et al. [KMV*98, KMA*00] proposed a medialness function which calculates a mean value of the gradient in a circle around the current position. The circle is oriented in a plane using eigenvectors from the Hessian matrix. In later work, they used a combination of the Hessian matrix and the structure tensor to extract the local orientation of structures in an image [KEV*03].

Pock et al. [Poc04, PBB05b, PBB05a, BPS*10] used two scale spaces for the gradient and Hessian in order to calculate a mean value of the gradient magnitude in a circle around the current position weighted by its contribution in radial direction. They took the symmetry of tubular structures into account by calculating a ratio between the medialness' variance and its mean. The final response is a combination of the described medialness function, and a vesselness measure calculated from the gradient magnitude.

Foruzan et al. [FZSH12] proposed an image feature to enhance the medial axis of tubular structures. Their idea is based on the fact that a point lies on the medial axis if the distances to the edges of the tube are equal. This is implemented as multiple lines which intersect the edges of the tube. They used the isotropic coefficient introduced by Pock et al. [PBB05b] to reduce responses for asymmetric cross-sections and dynamic thresholding to reduce noise sensitivity.

Geometric moments were used similar to Hessian-based approaches by calculating the eigenvectors of the second-order inertia matrix to estimate the local orientation of tubular structures [ELD10, CZP*10]. However, the computation of the inertia matrix is computationally costly. Laethen et al. [LJB10] used frequency-based quadrature filters to estimate the local phase along different directions.

Anisotropic diffusion The use of Gaussian blurring to create the scale space stack has the disadvantage that nearby structures may diffuse into each other. Some work [CR03, MVN06] modified the original edge-based anisotropic diffusion method by Perona and Malik [PM90]. Hereby, the diffusion process is guided by local anisotropy derived from the Hessian matrix and a vesselness measure. Metz et al. [MSvdG*07] used this as a preprocessing step to segment coronary arteries.

Bauer and Bischof [BB08] proposed to use the Gradient Vector Flow (GVF) as edge preserving anisotropic diffusion method to replace isotropic diffusion in linear scale space. Anisotropic diffusion schemes may require many parameters adjusted to a specific problem and are computationally very complex.

Morphological methods Another approach to the enhancement of vessels is the use of morphological operators. Rossant et al. [RBC*11] used them to preprocess retinal images. Concretely, they used the morphological top-hat operator with a disc-shaped structuring element slightly larger than the thickest vessel to remove background areas. Then they applied a morphological path-opening [HBT05] to remove noise at two scales.

2.2.2. Extraction

Using the preprocessed image, vessels are now segmented or centerlines extracted. Different classification schemes were proposed in literature to classify vessel segmentation methods. More generic classification schemes group segmentation methods in pixel-based, region-based, edge-based and model-based approaches. Lesage et al. [LABFL09] proposed three general classes, namely region growing-based, active contour-based and centerline-based approaches. This classification does not cover pixel-based or other energy-based methods. In this work, available extraction methods fall into one of the following classes:

- Pixel-based,
- Region-based,
- Energy-based and
- Centerline-based approaches.

2.2.2.1. Pixel-based

Global thresholding is probably the simplest pixel-based segmentation technique. It assumes that an object can be extracted from an image using a value that separates object from background. Therefore, a *threshold* $t \in \mathbb{R}$ with $min(I(x)) \leq t \leq max(I(x))$ is selected and a segmentation is calculated accordingly as

$$S(x) = \begin{cases} 1, & \text{if } I(x) \geq t \\ 0, & \text{else.} \end{cases} \tag{2.6}$$

The threshold parameter can be manually selected or automatically determined using prior knowledge. A correlation between the intensity value of a structure and its physical property may exist, such as, for example, in computed tomography. Thus, a threshold may be selected manually based on the structure to be segmented. To determine the threshold parameter automatically, the most popular way is to analyze the gray value distribution of an image in a preprocessing step using Otsu's method [Ots79] or using machine learning techniques such as statistical analysis (see Section 2.2.1.2). Proposed segmentation methods were used on preprocessed/vessel enhanced images in order to segment vessels from the liver [CFD*11, KZZ*07].

Threshold-based methods usually produce poor results if noise-levels are high or when the gray values of an object vary. To overcome the problem of varying gray values, local

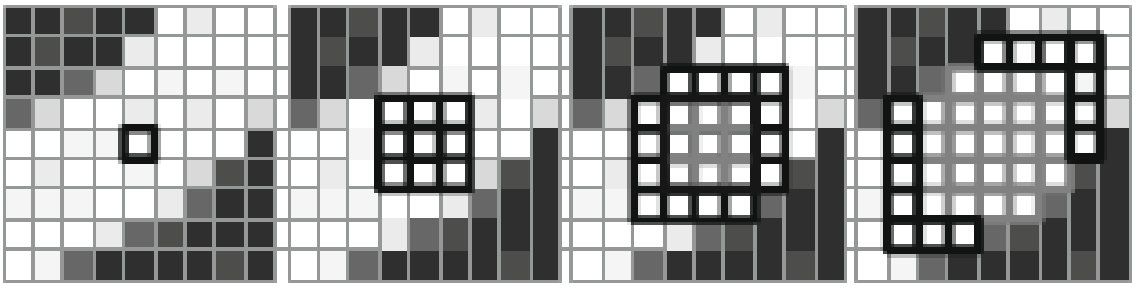


Figure 2.4.: Illustration of a region growing algorithm. From left to right: Initial seed point, region after one, two and three iterations. Image taken from [Sel99].

thresholding techniques can be used, where the threshold t is replaced by a spatial dependent variable $t(x)$. This was, for example, used by Wesarg et al. [WF04] to segment the coronary arteries.

Hysteresis thresholding is another technique to threshold images. It uses two thresholds T_{low} and T_{high}. Three types of voxels are distinguished. If $I(x) > T_{high}$ then it is called *strong*, if $I(x) \leq T_{low}$ *weak*. All other voxels are called *candidates*. A voxel that is weak is discarded and part of the segmentation if it is strong. If a candidate voxel is connected to a strong voxel via a chain of voxels with $I(x) > T_{low}$ it is also included in the segmentation, else discarded. Rossant et al. [RBC*11] used hysteresis thresholding to binarize vessels in retinal images.

General thresholding techniques are reviewed in the work by Sezgin et al. [SS04].

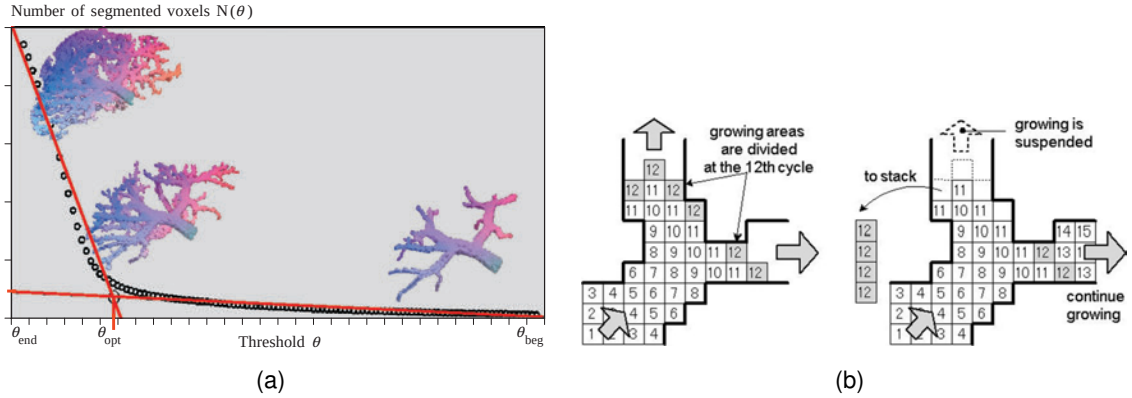
2.2.2.2. Region-based

Probably, the most popular region-based method is the region-growing algorithm. It assumes that voxels belonging to an object are connected and meet some similarity criteria. It exploits the idea that pixels, which are close together, have similar intensity values. This property makes region-growing methods in general more robust than threshold-based methods. The class of region growing methods can be subdivided into Seeded Region Growing and Unseeded Region Growing methods. Seeded region growing methods require seed points to mark objects. As illustrated in Figure 2.4, the region growing process starts at one of the provided seed points and adds neighbors to the region if a similarity criterion is met. Formally this is described as follow. Let O_0 be a set of user provided seed points inside an object and $N_\alpha(x)$ a set of voxels connected to x in a α-neighborhood with $\alpha = 6, 18, 26$ in 3D. Furthermore, let $H(x)$ be a similarity criterion. The region growing algorithm can then be recursively defined as

$$O_{n+1} = \{x | \exists o \in O_n . x \in N_\alpha(o) \wedge H(x)\}. \tag{2.7}$$

The recursion terminates when no more voxels are added to the set, denoted by $O_m = \{\emptyset\}$ for some $m \geq n+1$. The final segmentation is then given by

$$O = O_0 + O_1 + ... + O_{n+1} = \bigcup_{i=0}^{n+1} O_i. \tag{2.8}$$



(a) (b)

Figure 2.5.: (a) Automatic determination of a reasonable threshold by detecting a rapid slope change in a graph. Image taken from [SSPP00]. (b) Branch-based region growing. Image taken from [SSE*05].

Thus, a voxel is in O iff it belongs to the segmentation,

$$x \in O \Leftrightarrow S(x) = 1. \tag{2.9}$$

Because of their computational simplicity, region growing methods are very popular in the literature. Region growing methods differ mainly in the selected similarity criteria and image features used. For example, it can be based on thresholds or on mean and standard deviation of the pixels in the already grown region, but also on the output of vessel enhancing filters. It was used, for example, to segment vessels of the liver [BPJ*04, ERS08, LRV*08, SCG*08, SP09, CLV*10, NKM10, ZPFLSZ*11, OFC11], heart [MSvdG*07, BKBP08], brain [PRB*05, SSE*05], airways [THMS05, PLS09] and abdominal area [BMK10].

Some of these methods are also referred to as wave or front propagation techniques, e.g. [SSE*05, MSvdG*07, BKBP08]. Hereby, the region growing front evolves like a well-formed interface. Basically, this can be implemented by labeling voxels with the current recursion level (index n in Equation 2.7) instead of '0' or '1'. As shown later, this was used to detect leaks and bifurcations.

Selle et al. [Sel99] proposed a method to automatically suggest a reasonable threshold based on the observation that the number of segmented voxels is approximately linear for a threshold range where only vessels are segmented. The slope changes dramatically when the region grower starts to leak. In this case, a disproportionately high amount of liver parenchyma is segmented. This can be used to fit two regression lines to the two characteristic parts of the curve. The crossing of both lines is then suggested as the threshold. Figure 2.5(a) illustrates this idea. This method was later used in some other work [SSPP00, SPSP02, LRV*08, HZH*09].

Beichel et al. [BPJ*04] proposed a method which first determines an optimal threshold using Otsu's method on a vessel enhanced volume. It is used to calculate mean and standard deviation of foreground and background voxels. Seed points for both classes are then automatically placed using these statistical parameters. Foreground and background are then simultaneously grown using region growing. The similarity criterion is based

on the optimal threshold which is relaxed in each iteration step of the region growing algorithm.

Sekiguchi et al. [SSE*05] proposed a branch-based region growing algorithm with a dynamically adapted similarity criterion based on the average thickness of a branch. By evolving a region growing front, single branches are segmented and followed in a depth first search manner. The wave front propagation technique allowes for the detection of bifurcations and branches by counting connected components with respect to the voxel labels as illustrated in figure 2.5(b).

Erdt et al. [ERS08] automatically distributes seed points on local maxima of a vesselness filter output. The threshold-based similarity criterion for the region growing algorithm is iteratively lowered until 80% of the segmented voxels are above the mean intensity of the dataset.

Chi et al. [CLV*10] proposed a region growing algorithm that uses vessel context in their similarity criterion. In their work, vessel context is a feature vector that contains the intensity from the original image, the value from a vessel enhancing filter, the direction represented by the eigenvector corresponding to the largest eigenvalue of the Hessian matrix and the connectivity, which measures all potential paths to extend a tubular structure to another one.

Nimura et al. [NKM10] proposed a region growing algorithm that includes a vesselness measure and the eigenvector corresponding to the largest absolute eigenvalue in their similarity criterion. They found that segmentations can be improved where the vesselness measure is low when the eigenvectors are taken into account.

Zhan-Peng et al. [ZPFLSZ*11] developed a hierarchical region growing algorithm, which learns its similarity criterion automatically. Therefore, the image is divided into sub-blocks with a defined size, and the similarity criterion is estimated using statistical properties in the local regions around the seed points. It is calculated as the minimum difference between the gray level of a pixel and the average gray level of the assigned region. If a sub-block around a seed meets the similarity criterion, then it is included in the segmentation. Afterwards, each segmented sub-block is divided and examined again.

Oliveira et al. [OFC11] calculated a Gaussian mixture model of three Gaussians for liver vessels, nodules and parenchyma. Therefore, their method first calculates a histogram of the liver assuming that the parenchyma is dominant in it. Then, a Gaussian is fitted into this part of the histogram in a least-square sense using the Levenberg-Marquadt optimizer. This is then subtracted from the first histogram. The other two Gaussians are calculated in a similar way. An empirically determined threshold T_{high} is used to classify voxels directly as vessels which are used as seed points for a region growing algorithm. The region growing algorithm includes neighboring voxels as long as they are above a threshold T_{low}, which is derived from the mixture model.

Leak detection and prevention A disadvantage of classical region growing methods is that they tend to leak into neighboring structures. This problem was addressed by several researches [HTKM04b, THMS05, MSvdG*07, BKBP08, PLS09].

Heimann et al. [HTKM04b] proposed a solution based on the observation that leaks normally appear at narrow bottlenecks.

Metz et al. [MSvdG*07] proposed to prevent leaking into non-vascular structures by imposing a threshold on the number of voxels that may be added to the vessels every two iterations.

Bock et al. [BKBP08] proposed a similar solution by assuming that vessels get gradually narrower.

Tschirren et al. [THMS05] used adaptive cylinders (with respect to the radius) to bound the segmentation and proposed to prevent leaks by taking into account the observation that leaks appear at small holes at the border of the structure. They observed that leaks have a perforated (they called it 'spongy') structure containing holes and tunnels, which can be detected by topological thinning. Leaks are avoided by calculating a directional affinity as a function of the intensity and neighbors in the direction of the cylinder.

Pinho et al. [PSV06] improved this method by including anatomical knowledge like maximal number of bifurcations and decreasing diameters. In later work [PLS09], they further improved the idea by making the cylinders adaptive with respect to radius and height to detect more airway branches. Furthermore, their method avoids leaks by using a neighbour affinity, which tries to discover the size of the leaking hole.

Another disadvantage is that holes, bays and tunnels can appear in the segmentation. This is mainly due to noise and varying contrast. Appropriate preprocessing (Section 2.2.1) can reduce this effect significantly by removing noise or enhancing vessels, and a postprocessing step can fix the remaining problems (Section 2.2.3).

2.2.2.3. Energy-based techniques

This section outlines methods that are based on energies to achieve the segmentation goal. These methods are further subdivided into the following categories.

- Active Contours (including variational approaches),
- Graph cuts.

Active contours Active contours evolve an interface (the contour) through several energies or forces. External forces, derived from the image, pull the interface towards the boundaries of the object. Internal energies are based on the model and constraint the free moving space. A segmentation is found if external and internal forces are in equilibrium. This can be formulated as an energy minimization problem.

Parametric active contours [KWT88], also called *snakes*, rely on an explicit representation of the contour as parametric curve. Thus, a snake can be represented by

$$v(s) = (x(s), y(s)) \tag{2.10}$$

where $x(s)$ and $y(s)$ are coordinate functions and $s \in [0, 1]$. The total energy of the original snake is given by

$$E_{snake} = \int_0^1 (\alpha(s) \left|v_s(s)\right|^2 + \beta(s) \left|v_{ss}(s)\right|^2)/2 - \left|\nabla I(v(s))\right|^2 ds, \tag{2.11}$$

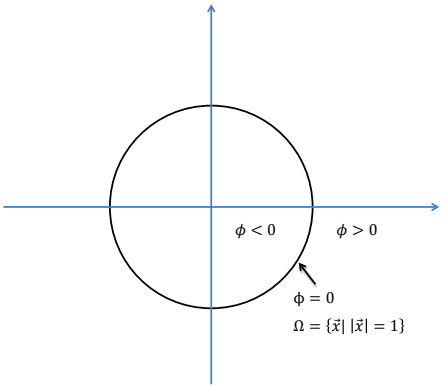


Figure 2.6.: Implicit representation of an interface and inside/outside regions.

where α and β are positive parameters to control the weight of each term. The first two terms are the internal energies which control the smoothness (elasticity and stiffness) of the contour. The third term is the external energy which pulls the contour towards the objects contour.

Xavier et al. [XOR*00] proposed a deformable model, called Eigensnake, to segment coronary arteries in 2D angiographies by replacing the original external energy with a statistical energy that describes a probabilistic model of the vessel.

Espona et al. [ECOP07] used a snake that incorporates domain-specific knowledge, such as topological properties of the vasculature, to segment retinal vessels.

Snakes can be efficiently implemented for 2D images. However, parametrization becomes more complex in 3D. Furthermore, changes in topology must be explicitly handled, and usually it is required to initialize the snake through user interaction.

Implicit active contours became popular by the use of the level set method, which adds dynamics to an interface. In contrast to parametric active contours, the interface is implicitly defined as the isocontour of some function ϕ. For example, the zero isocontour of $\phi(x) = x^2 + y^2 - 1$ is the set of points where $\phi(x) = 0$, which is the unit circle defined by $\Omega = \{\vec{x} \mid \|\vec{x}\| = 1\}$. Compared to an explicit representation it can also be easily evaluated, if a point is inside ($\phi < 0$) or outside ($\phi > 0$) the contour. This is depicted in Figure 2.6.

In order to define the motion of the contour the convection (advection) equation

$$\phi_t + \vec{V} \cdot \nabla\phi = \phi_t + u\phi_x + v\phi_y + w\phi_z = 0, \tag{2.12}$$

also referred to as the *level set equation*, can be used. In this partial differential equation (PDE) ϕ_t denotes the temporal partial derivative at time t and $\vec{V} = (u, v, w)$ is a velocity field, which pulls the contour towards the desired direction.

Level set methods were used to segment vessels of, for example, the liver [FP08, LJB10] and the lung [YMLB05].

Fei et al. [FP08] used the standard level set formulation (Equation 2.12) to segment liver vessels.

Laethen et al. [LJB10] formulated the liver vessel segmentation challenge as an energy minimization problem. The target function is set to the output of a quadrature vessel enhancement filter. The resulting energy formula is solved using the calculus of variation

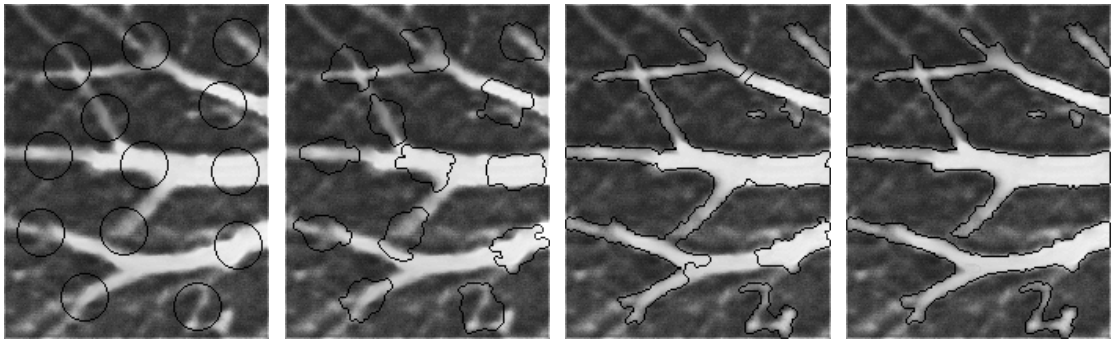


Figure 2.7.: Multiscale vessel segmentation using level sets. Image taken from [YMLB05].

and the Euler-Lagrange framework. A level-set PDE is then derived from the resulting curve evolution equation.

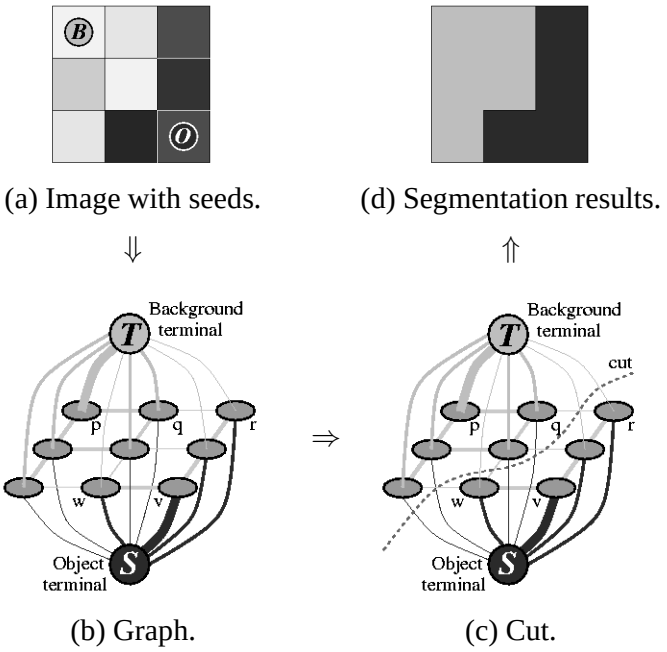
Yu et al. [YMLB05] used a Hessian-based vessel detection methods to incorporate the vessel shape prior into a region-based external force and combined it with geodesic active contours. The resulting speed function is embedded into the level-set framework to solve the segmentation problem. Figure 2.7 shows results applying this method to the human lung.

With discrete images, level set computations can be performed on a fixed Cartesian grid without the need for parametrization of the object. Furthermore, topology changes are implicitly handled. However, level set methods are in general computationally very demanding, and they must be properly initialized, which, in general, requires user interaction.

Graph cuts Many problems can be formulated in terms of energy minimization. Some of them can be reduced to a maximum flow problem in classical graph theory. For such a flow network, the max-flow/min-cut theorem states that the maximum flow capacity from source to sink equals the cost for a minimum cut through the graph in such a way that the flow from source to sink stops. Graph cut refers to methods, which employ a max-flow/min-cut optimization in order to find a global optimal solution with respect to an energy function. In case of image segmentation, a graph $G = (V, E, c)$ is constructed consisting of a set of nodes $V = P \cup \{s, t\}$, a set of edges $E = E_n \cup E_t$ and a cost function $c : V \times V \rightarrow R^+$ that defines the weight of the edges. The special nodes s and t define the object terminal node and the background terminal node respectively. E_n are neighborhood links (n-links) and E_t terminal links (t-links). The cost for a segmentation $S(x)$ can be calculated as

$$E(S(x)) = \lambda \cdot R(S(x)) + B(S(x)), \tag{2.13}$$

where $R(S(x))$ is a regional term that takes the probability of a voxel belonging to the object into account and λ a weight factor for this term. $B(S(x))$ is a term that takes the boundaries of an object into account. The problem is now to find a segmentation $S_o(x)$ such that $E(S_o(x))$ is minimal. $E(S(x))$ can be minimized by calculating a min-cut in a graph. The process from original image to segmented image is illustrated in



(a) Image with seeds. (d) Segmentation results.

(b) Graph. (c) Cut.

Figure 2.8.: 2D segmentation example using graph cuts. Image taken from [BJ01].

Figure 2.8. Very popular in literature is the max-flow/min-cut algorithm of Boykov and Kolmogorov [BK04].

Methods based on graph cuts were developed to segment vessels of the liver [HVN08, KTA09, ELD10, PWL11, PVWL12], retina [SGLL10] and heart [CC06].

Homann et al. [HVN08] used the conditional probability $P(I(x)|'Vessel')$ that a voxel belongs to a vessel as the regional term. The probability is derived from a probability intensity distribution that was constructed using interactively marked vessels.

Kaftan et al. [KTA09] used a highly selective vesselness measure to mark foreground voxels for a graph cuts segmentation. Hereby, the corresponding nodes are connected with infinity cost t-links to the source and data node, while the remaining nodes are connected with n-links.

Esneault et al. [ELD10] proposed a method that calculates the regional term using the probability density function of the intensity distribution of vessels using the output of a vessel enhancement filter. Furthermore, a third energy term is added to Equation 2.13. Therefore, strong information within detected vessels is diffused to its neighborhood along the vessel axis and constrained perpendicular to this axis according to the vessel radius.

Pamulapati et al. [PWL11, PVWL12] registered contrast enhanced and non-contrast enhanced CT data to define a regional term derived from three energy functions. A data energy computes penalties on a 4D histogram of object and background voxels. Another energy penalizes voxels that do not rapidly enhance between both phases, and the third energy is derived from a vesselness function.

A problem of graph cuts based methods is that memory consumption increases quickly as the image size increases. Recent work tried to overcome this problem by reducing the graphs before the computation is carried out [LML10]. Another approach is to use a multi-resolution scheme to create a rough segmentation in a low-resolution version of the image

which is interpreted as the region of interest in a higher-resolution version [DSOL11].

2.2.2.4. Centerline-based approaches

Centerline-based approaches differ from the previous extraction methods in the sense that they do not produce lumen information in the result. However, as described later (Section 2.2.3.3), the extracted centerlines can be used to constrain an accurate segmentation of the vessel lumen. Centerline-based methods can be further subdivided into local and global centerline-based methods.

Local centerline methods Local centerline methods rely on information provided by the local neighborhood of a voxel. Popular features are extracted from derivatives, e.g. Hessian and/or Gradient [AB02,BPS*10,YBE11] or statistical properties [AMR*10]. These usually start at one or more user-specified points within a vessel and from there try to track the centerline. Proposed methods also differ in the way bifurcations are handled. Depending on the application domain, only a single branch can be of interest [WF04]. Here, the algorithm does not detect branches at all and even tries to avoid them by starting at the end of a vessel and moving toward the beginning. Other methods try to detect bifurcations and continue following each branch independently [BPS*10]. Specifically, the liver has two densely distributed main vessel systems, which, due to partial volume effects, can appear as connected at some points. The separation of both vessel systems can be directly included in the algorithm by including model assumptions about both vessel systems [BPS*10].

Aylward et al. [AB02] proposed a method that starts from a seed point and follows intensity ridges. A ridge is defined using three conditions. First, the first two ordered eigenvalues of the Hessian matrix have to be both less than zero. Second, the projection of the gradient onto the corresponding eigenvectors must equal zero. Third, the ratio of the first two eigenvalues has to be greater or equal to one. Traversal directions are derived from the third eigenvector from the current and previous position. The optimal scale is dynamically determined every ten voxels.

Wesarg and Firle [WF04] developed a method to accurately extract the centerline of the coronary arteries based on user provided start, end, and direction points. Their method creates a rough estimation of the centerline by traversing the tube in a path formed like a helix (this gave their algorithm the name corkscrew algorithm) guided by a cost function. The center of gravity of three consecutive points is used as first estimate of the centerline. This estimate is then corrected by detecting the vessel's border in perpendicular direction to the centerline using equiangular rays originating from the current centerline voxel.

Bauer et al. [BPS*10] extracted centerline pieces from the response of a medialness filter using a height ridge traversal and hysteresis thresholding. Therefore, all local maxima above a threshold t_{high} are first extracted. Candidate points are then followed along their tangent direction as long as the medialness response stays above a threshold t_{low}.

Adel et al. [AMR*10] presented a method that iteratively tracks the vessel centerline by using edge points obtained from local statistical information about the intensity distribution, and information obtained in previous iterations. This algorithm is applied again if a bifurcation is detected.

You et al. [YBE11] proposed a method that performs a kernel interpolation of a vessel enhanced image and traces the principal curve using information from the Hessian, gradient and maximum response of a vesselness filter.

Local centerline methods are usually computationally efficient, because they focus on the object of interest and avoid unneeded calculations. However, they often depend on many parameters and have to face the problem of traversing accidentally in the wrong direction. Furthermore, they require robust termination criteria and the estimation of the next direction vector can become rather complex.

Global centerline methods Global centerline methods, also referred to as minimal path techniques, globally optimize the path from a start- to an end-point and are usually used to extract single branches of a vascular tree. The problem can be formulated as an optimization problem which minimizes a cost function and penalizes centerlines, which lie off center. Dijkstra's single source shortest path algorithm is a popular solution to this problem on a discrete grid and used in several works, e.g. [EMGF07, CJDL11]. Therefore, each voxel is interpreted as a node in a graph which is connected to its neighbors via edges and assigned costs. Dijkstra's algorithm finds the cheapest way from a start to an end node with respect to the assigned costs.

Egger et al. [EMGF07] used Dijkstra's algorithm with user provided start- and endpoints to find a shortest path with respect to costs calculated from intensity values. Afterwards, their method optimizes the alignment of the centerline within the blood vessel using an active contour model.

Chen et al. [CJDL11] presented a method that connects centerline pieces extracted from a multi-scale ridge detector using dijkstra's algorithm. Wrong bridges and branches were detected and removed using hypothesis testing.

The fast marching algorithm is an approach to solve for minimal paths in continuous space and frequently used in literature, e.g. [MM08, tBSSS09, BC11, KYT12]. It was introduced by James A. Sethian [Set01] as a numerical method for solving boundary value problems known as the Eikonal equation

$$F(x)\,|\nabla T(x)| = 1, \tag{2.14}$$

which describes the evolution of a point x of an interface as function of time T with speed F in the normal direction. Fast marching can be seen as a variant of implicit active contours, which evolves an interface only in outward ($\phi \geq 0$) direction. In this sense, it is very similar to Dijkstras's algorithm, which expands outward from a source node.

Müller et al. [MM08] improved the standard fast marching method for minimal path calculation by avoiding that the interface is attracted to places not belonging to vessels (e.g. calcifications and voxels outside the vessels). This is done by thresholding a vessel enhanced image by two thresholds for background and calcifications.

Brinke et al. [tBSSS09] used fast marching on a contrast enhanced image to calculate the centerline from the tip position of a catheter and a user-defined endpoint.

Li and Yezzi [LY07] improved the original fast marching technique by adding the vessel thickness as additional dimension to the search space. Later, Benmansour and Cohen [BC11] improved this idea by adding also vessel orientation derived from an optimally

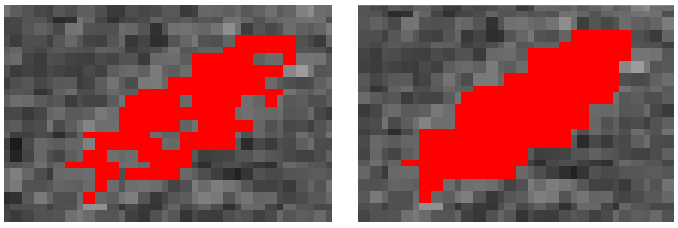


Figure 2.9.: Postprocessing using morphological closing to fill holes. Image taken from [CLV*10].

oriented flux descriptor to the model.

2.2.3. Postprocessing

Postprocessing is often necessary to obtain additional representations of the vessel system that are complementary to the previously generated one. For example, centerline-based extraction methods usually output only the centerline of the vessels. Thus, if also the lumen is required it must be created during a post-processing step. Another postprocessing task is to fix problems that were caused by the previous steps. These tasks can be grouped as

- Improvements,
- Skeletonization,
- Lumen segmentation,
- Graph creation,
- Tree reconstruction and separation, and
- Visualization

2.2.3.1. Improvements

After vessel extraction, several methods can be applied to improve the result. Usually, pixel-based methods also extract pixels that do not belong to vessels. A connected-component analysis can be performed to remove small spots in the image which do not belong to the vessels [KZZ*07, KTA09].

Region-based methods can result in segmentations that contain holes. This can occur, for example, if the contrast-agent is not optimally spread. Another cause is the enhancement of vessels during preprocessing. Vessel enhancement methods usually detect and enhance tubular structures. However, this assumption is violated at bifurcations, and fewer enhanced areas occur. Thus, postprocessing may improve the quality by closing those holes. Morphological closing is often used for this task in 2D [CLV*10] or 3D [BPJ*04]. Figure 2.9 shows an example of morphological closing in 2D. However, morphological closing can enhance the connection between different vessel trees. This can complicate the separation of both trees. Furthermore, morphological closing thickens the areas between bifurcations.

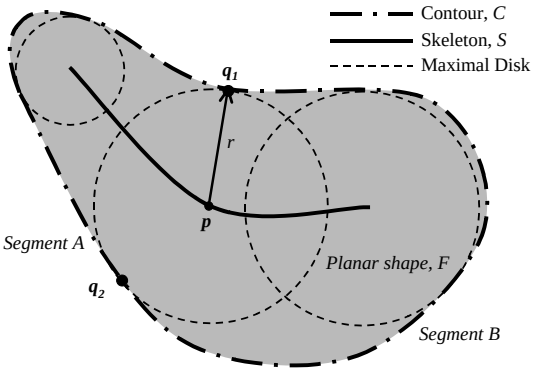


Figure 2.10.: Definition of a skeleton using maximal inscribed discs. Image taken from [CLS03].

2.2.3.2. Skeletonization (Segmentation->Skeleton)

The goal of skeletonization is the reduction of objects with dimension n to objects with dimension $n - m$, with $0 < m < n$. According to this definition, a 2D object can be reduced to an 1D object and a 3D object to a 2D or 1D object. Furthermore, a skeleton has to be topologically equivalent to the original object, lies geometrically in the center and is only one voxel wide.

Blum et al. [Blu67] compared the computation of skeletons with a prairie fire. According to his analogy, the surface of an object can be interpreted as grassland. If at every point on the surface, a fire is light up simultaneously that burns up with constant speed, then the skeleton is defined as the points where the fire fronts collide. Another definition is based on a distance measure. Let

$$S_0 = \{x | S(x) = 1 \wedge \exists y \in N_{26}(x).S(y) = 0\} \tag{2.15}$$

be a set of points on the surface of the object and $d(a, b)$ the distance between points a and b. Then

$$D = \underset{x \in S_0}{\operatorname{argmin}}\, d(x, y) \tag{2.16}$$

determines the shortest distance from a point to the surface. For some points there exist at least two surface points such that

$$D(x, S_0) = d(x, y) = d(x, z). \tag{2.17}$$

These points form the skeleton of the object. This can also be explained with maximal discs inscribed to an object. If a disc touches the surface in at least two points, then the center of the disc is a skeleton voxel.

There are several reasons why the computation of vessel skeletons is important. They enable guidance for systematic exploration of the entire tree and serve as path for navigation purposes in bronchoscopy [SH12]. Furthermore, skeletons pave the way for quantitative analysis of tubular structures, like vascular [SPSP02] or bronchial trees [PTHS06], where measurements must be obtained from cross-sections perpendicular to the long

axis. They were also used to detect plaques in computed tomographic angiographies [GK97, WKF06, WKJL08].

Several methods have been proposed in literature to calculate a skeleton for a given object. They are usually based on

- Voronoi diagrams,
- Distance transforms or
- Thinning.

Voronoi methods Methods based on Voronoi tesselation calculate the Voronoi diagram generated by the boundary points of an object; when the sampling rate goes to infinity, the corresponding diagram converges to the skeleton [OI92, IO92, ER02].

These methods satisfy both topological and geometrical requirements. However, it is a time-consuming process, especially for large objects. Therefore, these methods are unsuitable for volumetric medical images.

Distance transform methods Methods based on a distance transform convert the original image into a distance map where each element provides the distance to the nearest border. Ridges in the distance map are then interpreted as skeletal points.

Choi et al. [CLS03] proposed a connectivity criterion based on a set of point pairs along the object boundary, which are the closest contour points to the voxel under consideration in its 8-neighborhood. The idea is that a maximal disc can be inscribed into the object, and geometrical properties can be used to formulate the criterion. It can be used to decide if a single voxel is a skeletal point. Figure 2.10 illustrates this idea.

Chang et al. [Cha07] proposed a method that extracts ridge points based on the idea that the gradient at any point on a distance map generally points toward the ridge and reverses its direction as it crosses the ridge. A coarse graph is constructed from these points, and gaps are closed in a linking step. The graph is then converted to a set of snakes and deformed on the distance map to obtain smooth skeletons.

Latecki et al. [LnLByL07] proposed a new approach for skeletonization based on what they called skeleton strength map (SSM). The SSM is calculated using an isotropic diffused gradient vector field, which is obtained from the gradients of the distance map. Then, points on local maxima are selected and connected with geodesic paths.

Distance map based methods fulfill the geometrical requirement, but are sensible to disturbances produced by noise and generally do not guarantee the skeleton connectivity.

Thinning methods Thinning methods delete border points from the surface which satisfy certain conditions until a single point/voxel wide skeleton is left. Usually a 3x3x3 neighborhood for each border point is examined. An iteration can be further subdivided into subiterations, each responsible for the deletion of certain kinds of points. The concept of simple or deletable points plays an important role for thinning algorithms. A simple point is a point that can be deleted without changing the topology of the object. If an algorithm deletes these points one by one, they are called sequential thinning methods and parallel thinning methods, if they delete more than one per iteration.

Lee et. al. [LKC94] proposed a parallel thinning algorithm that repeatedly erodes deletable voxels until no more changes occur. They derived an Euler table to ensure the invariance of the Euler characteristic of the object during thinning and used quadtrees for neighborhood examination. The Euler characteristic was also used by Großkopf and Kieber [GK97] in their thinning algorithm.

Palágyi et al. developed several parallel thinning algorithms during the last two decades. Their algorithms mainly differ in the number of subiterations performed per iteration step. They developed algorithms with 6-subiterations [PK98, Pal01, PSB*01], 8-subiterations [PK98, PK99a] and 12-subiterations [PK99b]. They also applied their algorithm to analyze airways, blood vessels, and colons [PSB*01, PTS03, PTHS06].

Recently, Németh and Palágyi [NP11] compared 21 parallel thinning algorithms. They all produced different results, some with many unwanted side branches. A survey of thinning algorithms was written by Lam et al. [LLS92].

These methods preserve the topology and connectivity of the skeleton and guarantee the medial position of the skeleton, although the results are not necessarily in the completely accurate position.

2.2.3.3. Segmentation (Skeleton->Segmentation)

Centerline-based extraction schemes do not create a segmentation of the vessel lumen as explained in Section 2.2.2.4. Some work deals with the segmentation of the vessel lumen using the centerlines and some high-level information as input.

Wesarg et al. [WF04] used several rays that ran perpendicular to the centerline voxels to detect the vessel wall.

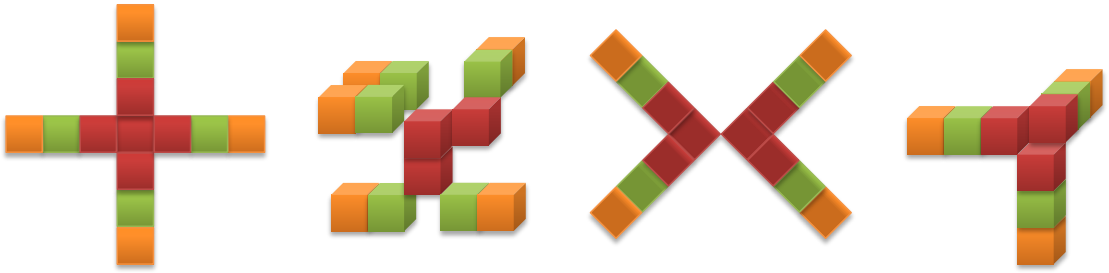
Kaftan et al. [KTA09] proposed a method that uses centerlines and a minimum radius estimate to mark foreground voxels for a following graph cut segmentation.

Bauer et al. [BPS*10] used the centerlines and estimated radii extracted from Hessian-based analysis to mark foreground and background voxels to segment the vessel lumen using graph cuts.

2.2.3.4. Graph creation

A formal graph representation of the vessel system has the advantage that further processing can be based on well-known graph theoretic algorithms. In medical imaging this can serve several purposes. For example, graph- or tree-matching algorithms can be used to find corresponding bifurcations, which serve as anatomical landmarks to guide a registration process [DOLCE10, OLD10, OLD11c]. Furthermore, graphs of interconnected vessel systems can be automatically analyzed with the goal to separate (see Section 2.2.3.5) or to annotate them [Sel99, PTS03, PTHS06]. They can also serve as the basis for an abstract visualization of the vessel system.

Graph creation methods can be subdivided into voxel-accurate and voxel-inaccurate methods. The idea behind voxel-accurate methods is that skeleton voxels can be classified based on the amount of neighboring voxels. If a voxel has only one neighbor, then it is classified as end-voxel. If it has exactly two neighbors, then it is classified as



(a) (b) (c) (d)

Figure 2.11.: Examples of ambiguous branch voxel classifications. Orange: End voxels. Green: Regular voxels. Red: Branch voxels. Images taken from [DOL10c].

regular-voxel, and if it has more than two neighbors, it is classified as branch-voxel. Finally, branch- and end-voxels represent nodes in a formal graph which are connected by edges.

The challenge for voxel-accurate methods is the proper classification of branch voxels, which is in many cases ambiguous, both in 2D and 3D as illustrated Figure 2.11. In these situations more than one voxel can be classified as branch-voxel (red voxels), but one has to be chosen to represent the branch in the formal graph.

Gerig et al. [GKS*93] created a graph that connects all adjacent vertices and then redundant edges are deleted based on a bit code for each voxel. The authors showed that this approach has problems with certain voxel configurations. Figure 2.11 (d) is an example that cannot be solved by their approach.

Lee et al. [LKC94] proposed to classify a voxel with three neighbors as "T" voxel and with four neighbors as "X" voxel. Any adjacent voxel is then considered as regular voxel. The authors showed that there are cases where this scheme produces undesired results. Figure 2.11 (a) is an example of where this approach will cause the ramification to be split into two bifurcations.

Selle [Sel99] solved for ambiguous situations by removing redundant branch voxels. Because this did not solve all problems, the author introduced an additional optimization metric. In the end, there were still ambiguous situations. For example, when three branch voxels are pairwise neighbors (e.g. Figure 2.11 (b), (d)). Some of these problems were solved by applying heuristics, which causes different connections depending on the search direction. Others are unsolved and left for later processing steps.

Palágyi et al. [PTHS06] selected the voxel closest to the root to solve for ambiguous situations. However, the root has to be manually selected in a previous step. If there are two interconnected vessel systems, like in the liver, then this method does not work, because there are two roots present.

Chen et al. [COLD09] applied a cost function to each branch voxel to find the branch voxel with the highest connectivity. Therefore, for each branch voxel each connection to neighboring voxels is evaluated. Face-connections are assigned higher costs than edge-connections and edge-connections are assigned higher costs than vertex-connections. The voxels with the highest cost are selected as final branch voxel. However, symmetric situations can still occur, which cannot be resolved as shown in Figure 2.11 (c), (d).

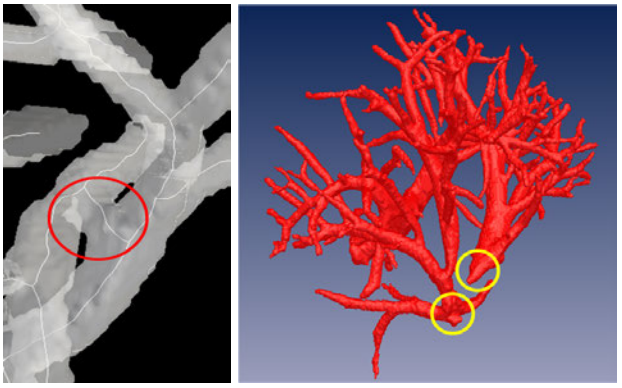


Figure 2.12.: Segmentation imperfections. Left: Interconnected branches of different vessel systems. Right: Splitted Vessel tree. Images taken from [DOOL11].

Voxel-accurate graph methods are accurate representations of the underlying skeleton. They do not loose branch-voxels of the skeleton in the graph representation, but impose problems on the selection of a proper branch-voxel.

Voxel-inaccurate methods also classify voxels as end, regular and branch voxels. The main difference is how branches are treated. Instead of choosing one branch voxel as representant of the branch, all potential branch voxels in a neighborhood are taken into account. Klette [Kle06], for example, used the centroid of the branch voxel cluster as representant of the branch.

Voxel-inaccurate methods cause an inaccuracy between branch node in the graph representation and branch-voxel in the skeleton. There is not a one-to-one mapping of both representations. Furthermore, visualizations of the skeleton can degrade and do not fit to the segmented vessels.

2.2.3.5. Tree reconstruction and separation

The liver contains several vessel systems, namely the hepatic vein system, the portal vein system and the liver artery system. The latter usually follows very closely the portal vein and is very thin, while the former two are much thicker and important for computer assisted surgical planning. The proper separation of both vessel systems is crucial for discriminative visualization during surgical planning, automated analysis (e.g. which vessel system is connected/closest to a tumor) and also for liver segment approximation, which can be computed using the Portal Vein [SPSP02, BPJ*04]. However, due to low resolutions, partial volume effects, unequal distributed contrast agent, motion artifacts and imperfect segmentation algorithms it happens that portal vein and liver vein systems appear to be connected at some points (e.g. Figure 2.12 left). Furthermore, a vessel system can be split into multiple parts (Figure 2.12 right). Separation methods can be classified as either voxel-based or graph-based methods.

Voxel-based methods Voxel-based methods try to separate vessels by classifying voxels to belong either to the portal- or hepatic vein.

Shang et al. [SCG*08, Sha10] assumed that only the portal vein is visible in the arterial phase, while both, hepatic- and portal veins, are visible in the venous phase. Their proposed method registers images of both phases and, after binearization of the vessels, subtracts them to remove the portal vein from the venous phase CT image.

Chi et al. [CLV*10] used a feature vector, which they call *vessel context*, consisting of intensity, saliency, direction and connectivity to group voxels belonging together into branches. These branches are then reconnected to trees using a context-based voting mechanism.

Sboarina et al. [SFM*10] proposed to create a rough classification of both vessel systems by means of two competing region growing algorithms starting at the root of each vessel system. Hereby, the user has to adjust parameters manually until both trees are roughly separated. Then a morphological opening is performed to remove remaining connections. Skeletons of the resulting vessel systems are generated and grown by classifying previously unclassified voxels connected to one of the already classified vessel systems. Finally, unclassified voxels connected to both vessel systems are classified by taking into account that abrupt changes in direction do usually not appear.

Pamulapati et al. [PWL11] proposed to process segmented vessels by morphological opening then to obtain the main branches of both vessel systems followed by a competitive region growing to restore small branches.

Oliveira et al. [OFC11] proposed a method that classifies branches to belong to the hepatic vein if they were longer than 15% of the liver height. Therefore, the biggest component in the first slice containing vessels is searched and merged with the biggest overlapping component on the next slice until no overlapping vessels are found anymore. This is repeated until three main branches are found. The portal vein is identified as the biggest connected component when the previously found hepatic veins are subtracted from the vessel segmentation.

Graph-based methods Selle et al. [SPSP02] proposed an automatic graph-based method to separate hepatic vessels. They assumed that hepatic vessels (a) are tree-shaped, (b) do not contain loops and (c) the thickness decreases in flow direction. First, their algorithm automatically detects a root node as follows. First, the node with the thickest diameter is searched. Second, all nodes which are bigger than 90% of this diameter are determined. Third, the node which is the farthest from the center of gravity of all skeleton voxels is chosen as the root node. The separation algorithm starts with edges that are outgoing from this node and appends new edges successively to the tree. For an appended edge, the ratio between its diameter and the diameter of the thickest remaining edge is calculated. If it is above a threshold then a new vessel system is detected, and the whole process starts again.

Göpfert et al. [GGD*98] proposed a semi-automatic method to separate hepatic vessels based on branch diameters. They assume that erroneous branches can only occur on paths between root nodes of different vessel systems and remove branches with the smallest diameter on these paths.

Homann et al. [HVN08] proposed a semi-automatic method to separate hepatic vessels. Similar to Selle et al., they used a-priori knowledge of the hepatic vessel system.

In addition to Selle's assumption, they assumed that hepatic vessels contain no obtuse angles at branches. Their algorithm appends edges to the portal vein as long as the diameter and angle criteria are met. Then the process is repeated for the hepatic vein.

Bauer et al. [BPS*10] proposed an automatic method to separate hepatic vessels. Their method extracts local maxima from a multi-scale medialness filter and reconnects tube elements based on a confidence function to preserve flow direction among the trees. Similar to the aforementioned authors, they assume that hepatic vessels have decreasing radii and the direction does not change abruptly.

Pamulapati et al. [PVWL12] proposed a graph-based method to identify valid subtrees in an undirected graph representation of the segmented vessels. They assume that the diameter must progressively increase from a child node towards the root of the tree. Furthermore, the minimal angle between a child node and a parent node across a branching node must be obtuse.

2.2.3.6. Visualization

Another aspect, which is not subject of research in this work, is the visualization of vessels. Marching Cubes [LC87] is probably the most popular method to construct surfaces from segmented volumes. However, it can produce bumpy meshes due to noise and artifacts. This problem was, for example, addressed by Hahn et al. [HPSP01]. Their proposed method fits cones with locally estimated radii to a vessel path. The diameters can be interactively smoothed by the user to create an appealing visualization.

Another problem is a large number of triangles that are created, which causes slow rendering. To overcome this, Yuan et al. [YCHL12] proposed a method to model n-furcated liver vessels from 3D segmented volumes. It consists of several steps. Centerlines are created and a graph is constructed where each centerline voxel corresponds to a node. The graph is converted to a tree by adopting a minimal path strategy to remove cycles. Then, the tree is broken up in separate branches and circles are fitted to the vessel cross sections and meshes are constructed. Finally, all meshes are combined by making holes at bifurcations where pieces are attached together.

Existing methods usually assume a circular cross section. However, the accuracy of small vessels can be negatively affected this way. Kretschmer et al. [KBT*12] proposed a method that models vascular structures with non-circular cross sections, while keeping the number of triangles low. In their method, vessels are decomposed into segments, which are locally described by implicit functions and combined using boolean operations. The surface is generated by adapting a bisection method to determine iso-surface positions. An octree data structure is utilized, to efficiently construct the mesh with a locally adapted triangle size.

2.2.4. Validation

One very important aspect of every vessel extraction method is how good it was tested and under which circumstances. This can increase the confidence in a specific method and actually, a decisive factor to use it. A plethora of possibilities exists to analyze the performance of an algorithm, but recent review papers [KQ04, LABFL09] neglected this

aspect and did not include a taxonomy for validation methods to classify available algorithms accordingly. This section fills this gap.

Two aspects are important when evaluating vessel segmentation algorithms: *data creation* and *evaluation measures*. Both aspects are described in the following sections.

2.2.4.1. Data creation

In general, the creation of data used for evaluation consists of two steps:

1. Generation of synthetic or simulated ground-truth data
2. Perturbation of the ground-truth data to simulate effects of the acquisition process to create test datasets

Figure 2.13 provides an overview of possibilities to create ground-truth data for validation of vessel segmentation algorithms. Representations of realistic situations under study have to be created in the first step. Three ways of doing this are conceivable.

Synthetic data Synthetic data can be created that do not represent realistic vessel systems. This can be done virtually [AB02,LJB08,FJS09,LJB10,NKM10] on the computer or physically as phantom models [BPS*10, CFD*11].

Synthetic tubes with gaussian profile can be generated by variations of the function

$$I(x, y, z) = e^{\frac{-(x^2+y^2)}{2\sigma^2}}. \tag{2.18}$$

Figure 2.14 shows some examples of public available[1] synthetic virtual data (SYVD) which represent different situations. This can be useful to study a specific aspect of an algorithm, e.g. how good it behaves at junctions. Aylward et al. [AB02] used SYVD to quantify parameter estimates of their centerline-based extraction method. Freiman et al. [FJS09] used SYVD for qualitative comparison of different algorithms at critical locations, e.g. junctions (see Figure 2.14). Laethen et al. [LJB10] used SYVD to quantify errors produced by different parameter choices and for qualitative illustration of results. Nimura et al. [NKM10] quantitatively compared their method with another algorithm based on SYVD.

Figure 2.15 shows examples of synthetic physical models (SYPM). Bauer et al. [BPS*10] used the right phantom to quantitatively evaluate the performance of their graph cuts based extraction method under varying physical conditions. The left SYPM was used by Conversano et al. [CFD*11] to quantitatively evaluate their method.

Synthetic data are useful to demonstrate the behavior of an algorithm or a system in an artificial environment. However, they are usually not intended to describe real patient data realistically and hence do not provide practice-related information about the performance of an algorithm.

[1] http://lmi.bwh.harvard.edu/research/vascular/SyntheticVessels/SyntheticVesselImages.html

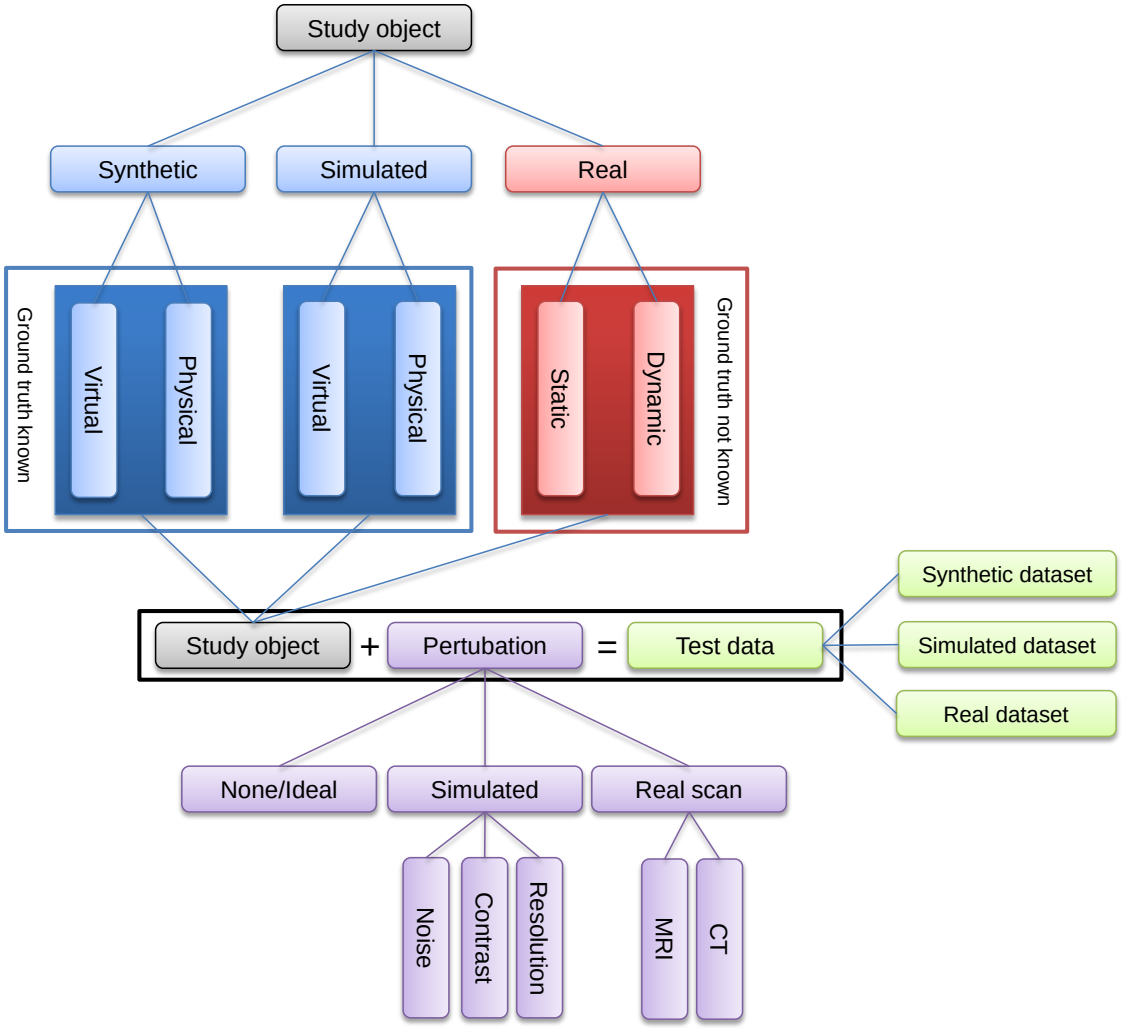


Figure 2.13.: Illustration of different methods to create test data for evaluation purposes. At the beginning a study object (grey) must be chosen. Study objects in blue represent synthetic or simulated objects where all relevant parameters are known. They can be directly used as ground-truth to compare with segmentation results. Objects in red represent objects (e.g. exanimated organs or living beings) where no ground-truth is known a priori. A manual expert segmentation is required for quantitative evaluation. Perturbations (purple) can be added to produce a test dataset (green). Perturbations can be either simulated or are added by the scan process using a physical scanner device, like CT or MRI. The former has the advantage that all parameters are known, while the latter produces more realistic test data.

Simulated data Simulated data improve upon synthetic data by means of realism. Computer generated vessel trees try to imitate real data as good as possible. The most wide-spread methods to modeling vascular systems algorithmically are mainly based on Lindenmayer systems (L-systems) or on physiological principles and constrained optimization [KSNN98, KTS99, KNNS99, BWKR03, KRBWC03, HGP05].

Figure 2.16 shows simulated virtual data (SIVD) of the human airway tree and portal

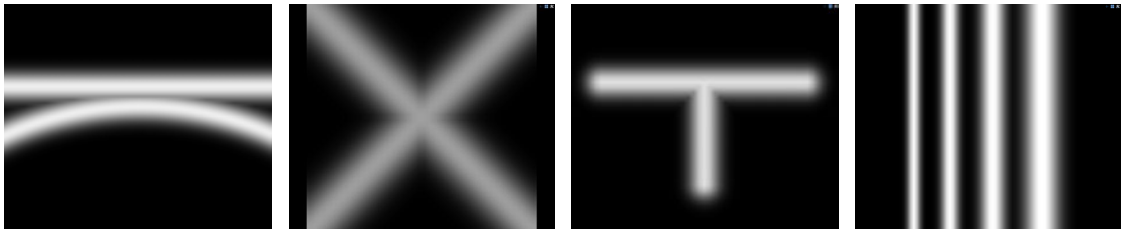


Figure 2.14.: Examples of synthetic virtual data representing special situations. From left to right: Nearby tubes, cross junction, T-junction, varying diameters.

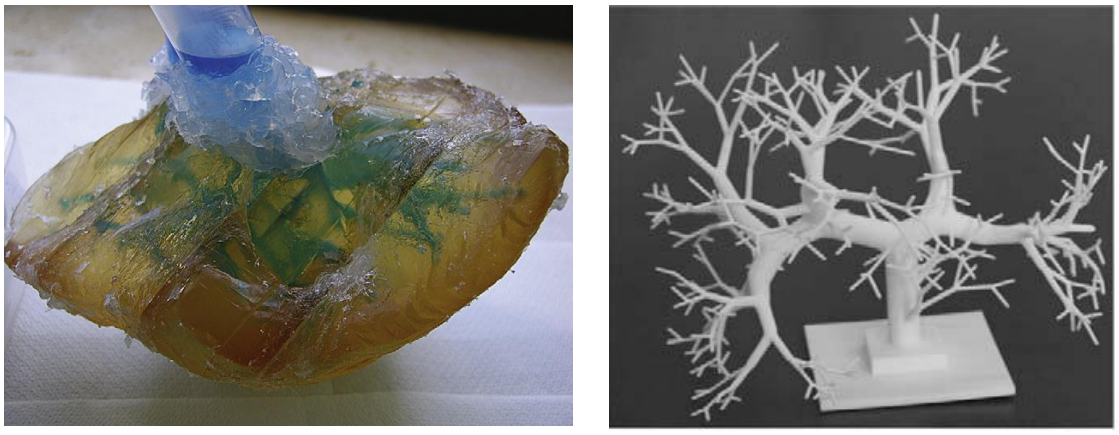


Figure 2.15.: Examples of synthetic physical data. Left: In gelatin embedded tubes. Image taken from [CFD*11]. Right: Plastic model of connected tubes with decreasing diameter. Image taken from [BPS*10].

vein of the liver. Both SIVD were created using physiological principles. Collins et al. [CKL*11] proposed an analysis framework to evaluate the quality and accuracy of vessel segmentation algorithms using such a SIVD as ground-truth.

Computer simulated data are complex to generate and can only be as realistic as the simulation allows for. However, they reflect real patient data much better than synthetic data. Furthermore, full control of all simulation parameters results in ground-truth data with exactly known properties.

Real data In terms of realism, real data cannot be beaten. They reflect a broad range of anatomical variations and anomalies that are usually not included in synthetic or simulated data. Examples are varying branching patterns, changing curvature, variable diameters of a single branch, varying contrast-agent spread and aneurysm to name a few. Real data can be based on dynamic or static object.

Real dynamic data (REDD) are typically clinical datasets. They carry the attribute *dynamic*, because breathing and organ motion, patient movement and blood flow are additional sources of error during the acquisition process. For quantitative evaluation, ground-truth data must be obtained by manual segmentation, ideally by an expert.

Real static data (RESD) avoid these additional sources of error by using exanimated organs or corrosion casts. Examples are shown Figure 2.17. Beichel et al. [BPJ*04]

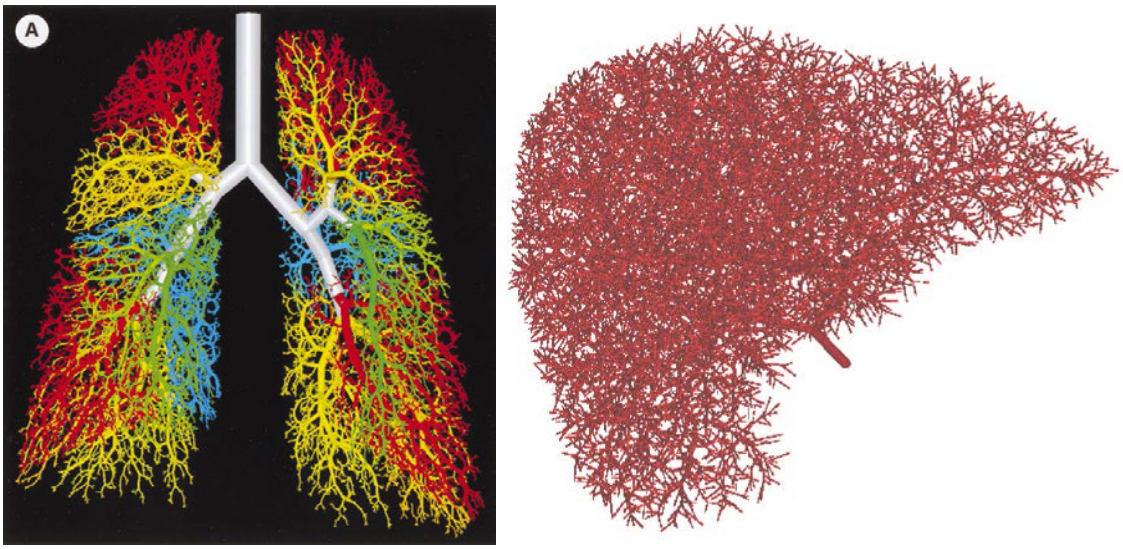


Figure 2.16.: Examples of simulated virtual data. Left: Computer generated vessels of the lung. Image taken from [KTS99]. Right: Portal vein of the liver. Image taken from [BWKR03].

Figure 2.17.: Examples of real static data. Left: Artificially perfused porcine liver. Image taken from [PDB*09]. Right: Corrosion cast of a porcine liver. Image taken from [Sel99].

used a corrosion cast model to quantitatively compare their method Hessian-based region growing method to a classical region growing algorithm. Lehmann et al. [LRV*08] used a corrosion cast model of a porcine liver similar to the one shown in the right image of Figure 2.17 to evaluate their segmentation algorithm applied to the same porcine liver extracted from an in-vivo CT scan of the pig. Peterhans et al. [PDB*09] used an artificially perfused porcine liver for a vessel-based registration of ultrasound and CT. Perfusion was necessary to prevent collapsing of the vessels (Figure 2.17 left).

Real dynamic data should be included in every serious evaluation to show the usefulness of a proposed method in clinical practice. However, due to too many non-controllable parameters, quantitative evaluation is complicated. Ground-truth segmentations must be

manually obtained and are subject to interobserver variability. This and different data used for evaluation make objective comparisons of results from distinct groups difficult. Real static data eliminate most of the sources of error. However, creating physical models is expensive and time-consuming. Corrosion casts have to be done by experts, which is not a 'service' available to everyone. Exanimated organs must be obtained from the local butcher and can only be used for a very short time until they molder. The main disadvantage of real data is that exact properties are not known, e.g. varying diameters. Some can be measured, but this is an excessively elaborate process and involves image processing, which can introduce errors.

Scan process After an appropriate data model has been chosen, it can be further processed to simulate the clinical image acquisition process. If a physical or static model has been chosen, it can be scanned by an authentic medical device, such as a computed tomography (CT) or magnetic resonance imaging (MRI) scanner. To scan an object under different controlled conditions requires access to such a device. A virtual model can be disturbed by, for example, noise, shadows, deformations and varying contrast to simulate clinical acquisition conditions.

2.2.4.2. Quantitative measures

In the previous section different ways of creating ground-truth data were explained. To quantitatively compare two segmentations, a variety of measures exist. They can be voxel-based or based on the structure of the vessels. Basically, they can be grouped into

- Statistical measures,
- Overlapping measures,
- Distance measures or
- Vessel-specific measures.

Statistical measures If a result can be binary classified based on whether it meets some criteria, which is the case for segmentation algorithms, then the terms *true positives* (tp), *true negatives* (tn), *false positives* (fp) and *false negatives* (fn) refer to a comparison of the obtained result with a trusted reference result. The terms *positives* and *negatives* hereby refer to the obtained result (also known as observation), and the terms *true* and *false* refer to the expected result (ground-truth). Table 2.2 illustrates all possible combinations. True positives are correct results, while true negatives correctly not belong to the result. False positives are results, which are not present in the ground-truth data, and false negatives refer to missing results, although they are present in the reference data.

Sensitivity and specificity are statistical measures to evaluate the outcome of a binary classifier. Sensitivity measures the proportion of true positives, which were correctly classified and is defined as

$$Sensitivity = \frac{tp}{tp + fn}. \tag{2.19}$$

	actual class (expectation)	
predicted class (observation)	true positive	false positive
	false negative	true negative

Table 2.2.: Combinations of expectation (true/false) and prediction (positive/negative).

Specificity measures the proportion of true negatives which were correctly classified and is defined as

$$Specificity = \frac{tn}{tn + fp}. \tag{2.20}$$

A perfect segmentation algorithm would be close to 100% sensitivity and specificity.

The positive predictive value (PPV), also called precision, measures the proportion of true positives against all positive results. It is defined as

$$PPV = \frac{tp}{tp + fp}. \tag{2.21}$$

The negative predictive value (NPV) measures the proportion of true negatives against all negative results. It is defined as

$$NPV = \frac{tn}{tn + fn} \tag{2.22}$$

Accuracy is the proportion of true results against all results. It measures how well a binary classifier includes or excludes the expected result. It is defined as

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn}. \tag{2.23}$$

Chi et al. [CLV*10], for example, used Sensitivity, Precision, Specificity and Accuracy to compare their algorithm to segmentations performed by a medical expert.

Overlapping measures A common performance measure to compare two segmentations is the *Dice coefficient*. Let S_0 and S_1 two sets of segmented voxels, then

$$C_{Dice} = \frac{2\,|S_0 \cap S_1|}{|S_0| + |S_1|} \tag{2.24}$$

measures the overlap of two segmentations. A similar measure is given by the *Jaccard similarity coefficient* given by

$$C_{Jaccard} = \frac{|S_0 \cap S_1|}{|S_0 \cup S_1|}. \tag{2.25}$$

Both measures evaluate close to 1 if the segmentations are very similar. They mainly differ in how the overlapping areas are normalized.

Overlapping measures were for example used in [CFD*11, NKM10, CLV*10, HVN08].

Distance measures The Hausdorff distance quantifies the maximal distance between the segmented surfaces. It is defined as

$$H(S_0, S_1) = \max\{d_{max}(S_0, S_1), d_{max}(S_1, S_0)\}, \tag{2.26}$$

where

$$d_{max}(S_0, S_1) = \max_{s \in S_0} \min_{t \in S_1} \|s - t\|. \tag{2.27}$$

Distance measures were for example used in [LJB10, ELD10, CLV*10].

Vessel specific measures To measure the performance of vessel segmentation algorithms, specific measures based on structure, topology and properties of a vessel system can be used.

Beichel et al. [BPJ*04] proposed to detect missing vessels by subtracting found vessels from ground-truth data and counting connected components. They were then analyzed regarding their lengths and radii. Freimann et al. [FJS09] counted the number of bifurcations to compare the results from two extraction methods. Bauer et al. [BPS*10] measured the performance of their algorithm by counting the number of detected tubes and the relative diameter error.

Statistical measure can be also combined with vessel specific measures. Lehmann et al. [LRV*08] measured sensitivity of the segmented vessel diameters and precision of the segmented vessel orders. Conversano et al. [CFD*11] measured sensitivity of segmented vessel diameters and of vessel order.

2.2.5. Summary

In the previous sections, current state of the art methods have been reviewed. Although the focus was directed towards hepatic vein extraction, a plethora of algorithms for other organs has been also included. Table 2.3 summarizes some of the hepatic vein extraction methods according to preprocessing, extraction and postprocessing.

Table 2.4 summarizes the used validation methodology to evaluate vessel extraction algorithms.

2.3. Discussion

This section discusses available hepatic vein extraction methods and shows their strengths and weaknesses.

Preprocessing Vessel enhancement as a preprocessing step dominates in available algorithms. Especially derivative-based methods are very popular. They can be combined with pixel-, region- and energy-based methods to improve the segmentation result, but they are also used in centerline-based methods to extract the centerline. Compared to anisotropic diffusion schemes, they are faster to calculate and do not only enhance vessels, but also suppress non-vessel areas and need fewer parameters. When compared

Publication	Preprocessing			Extraction	Postprocessing				
	Misc	Smoothing	Vessel Enh.		Corrections	Skeletonization	Segmentation	Graph	Tree sep.
Pamulapati2011	Registration		Derivative	Energy		Thinning			Voxel
Conversano2011			Derivative	Pixel					
Nimura2010			Derivative	Region					
Laethen2010			?	Energy					
Esneault2010			Integrative	Energy					
Chi2010			Derivative	Region	Holes				Voxel
Bauer2010			Derivative	Centerline			Energy	Accurate	Graph
Seo2009			Derivative	Region	Noise				
Kaftan2009	Lesion removal	Mean-Gaussian	Anisotropic	Energy	CCA	Thinning	Energy		
Freimann2009			Derivative	Energy					
Shang2008	Registration	Median	Derivative	Region		Thinning			Voxel
Lehmann2008		N/A		Region		Thinning			
Hofmann2008		Anisotropic		Energy		Thinning		Accurate	Graph
Fei2008				Energy					
Erdt2008			Derivative	Region					
Kawajiri2007		Median	Derivative	Pixel	CCA				
Beichel2004			Derivative	Region	Holes	Thinning		Accurate	
Aylward2002a				Centerline					
Selle1999		Median		Region		Thinning		Accurate	Graph

Table 2.3.: Summary of major publications dealing with hepatic vein extraction.

Publication	Ground-truth			Scan			Evaluation methodology				
	Synthetic	Simulated	Real	None	Simulated	Real	Overlapping	Distance	Vessel specific	Qualitative	Comparison
Pamulapati2011			d							•	
Conversano2011	p					•	•		•		
Nimura2010	v		d		•		•			•	•
Laethen2010	v		d		•			•		•	•
Esneault2010	v		d		•		•	•		•	•
Chi2010			d*				•	•		•	
Bauer2010	p		d			•			•	•	•
Seo2009			d							•	
Kaftan2009			d							•	
Freimann2009			d*						•	•	•
Shang2008			d							•	
Lehmann2008			s						•		
Homann2008			d*				•				
Fei2008			d							•	
Erdt2008			d							•	•
Kawajiri2007			d							•	
Beichel2004			s						•		
Aylward2002a	v		d		•				•	•	

Table 2.4.: Summary of validation methodology of the publications mentioned in Table 2.3. Legend: p:physical, v:virtual, s:static, d:dynamic, d*:dynamic with manual ground-truth segmentation. Scan is specified with respect to synthetic or simulated ground-truth data.

	Morphological	Derivative	Anisotropic diffusion
Performance	0	0	-
Vessel enhancement	-	+	+
Background suppression	0	+	-
Parameters	+	+	-

Table 2.5.: Properties of different vessel enhancement methods.

	Pixel	Region	Energy	Centerline
Performance	+	+	0	0
Quality	-	0	+	N/A
Initialization	+	+	-	+

Table 2.6.: Properties of extraction methods. Centerline-based methods do not produce a segmentation of the vessel lumen and are therefore not comparable to the other methods regarding their segmentation quality.

to morphological methods they provide better vessel enhancement and background suppression abilities with the same processing speed. Table 2.5 summarizes properties of different methods.

Extraction The previous literature review has shown that the output of a vessel enhancement filter can be combined with all available extraction schemes to improve the outcome. Pixel-based methods are simple to implement, but lack segmentation quality. Energy-based methods are computationally expensive and require some effort for proper initialization. Centerline-based methods do not segment the vessel lumen, which has to be done in a postprocessing step using sophisticated methods. This goes at the expense of runtime performance, making this method the most expensive one. An advantage of centerline-based methods could be that the separation of interconnected vessels can be integrated in the traversal process. In practice, this process can be erroneous and user interaction may be required. However, a high degree of integration between centerline tracking and vessel separation makes user-friendly interaction during the execution of a centerline-based algorithm difficult. It seems reasonable that user-interaction is provided in some way in a post-processing step. However, then it makes no difference if the vessel separation is completely done in a post-processing step. Thus, a high degree of integration is not an advantage in case of liver vessel extraction. Region-based methods seem to be the best compromise between performance and quality. Table 2.6 summarizes the properties of different extraction schemes.

Postprocessing As already explained before, the liver contains two densely distributed veins, which can be touching at some points. Separation of both vessel systems can be performed on a voxel-basis or graph-based. Voxel-based methods are usually computationally expensive. Some methods require the registration of different phases or several classification steps. It is also difficult to interact with these algorithms, in case the separation results contain errors. Graph-based methods rely on well-known graph theoretic algorithms and are efficient for graphs of the size of a typical vessel system. Furthermore,

	Voxel-based	Graph-based
Performance	-	+
User Interaction	-	+

Table 2.7.: Properties of voxel- and graph-based vessel separation methods.

	Voronoi	Distance transform	Thinning
Performance	-	+	+
Topology	+	-	+
Geometric	+	+	+

Table 2.8.: Properties of skeletonization methods.

graphs (edges and nodes) can be visualized such that a user can interact with them to perform corrections manually. Table 2.7 summarizes the properties of both separation methods.

Graph-based separation methods require that a previously generated segmentation is transformed into a formal graph representation. Therefore, segmentations are first transformed into an intermediate representation; the skeleton. Voronoi methods meet topological and geometrical requirements, but are very expensive operations, especially for medical imaging data. Methods based on distance transforms fulfill geometric requirements, but in general do not guarantee the topological properties. Thinning methods seem to be the best choice as they force skeletons to be in the geometrical center, preserve topology and can be efficiently implemented. Table 2.8 summarizes the properties of the discussed methods.

The next step is the transformation of the skeleton into a formal graph representation. Voxel-accurate methods take every single voxel into account and thus have voxel accuracy, and no information is lost. They can be used in discrete and continuous space. However, they are complex to create, because the traversal of the skeleton in branch areas is difficult due to ambiguous situations. On the other side, voxel-inaccurate methods are easy to create, but can only be used in continuous space and are somewhat inaccurate at bifurcations. Table 2.9 summarizes the properties of both methods.

Validation As presented in section 2.2.4 a variety of possibilities exist to evaluate a vessel segmentation algorithm. However, most of the reviewed papers included their own validation methodology, which makes the comparison between different reported results difficult. A few synthetic datasets are public available, but they have almost no practice-

	Voxel-accurate	Voxel-inaccurate
Accuracy	+	-
Complexity	-	+
Visualization	+	0
Discrete	+	-
Continuous	+	+

Table 2.9.: Properties of graph creation methods.

	Meaningful	Realism	Inter-/Intraobserver variability
Synthetic data	-	-	+
Simulated data	+	0	+
Real data	0	+	-

Table 2.10.: Properties of data used for validating segmentation algorithms.

related value, although they can represent some specialized situations. Results produced with synthetic data are not meaningful with respect to transferability of quantitative results to real data. Simulated data imitates real data in a very controlled way up to a point where it is hard to distinguish them from clinical datasets. It is not prone to inter- and intraobserver variability. Quantitative results produced with simulated data are therefore meaningful. Using real data is subject to inter- and intraobserver variability due to the manual creation of ground-truth segmentations. Furthermore, the generation of real data cannot be controlled very well (e.g. motion and breathing artifacts). Therefore, quantitative results produced with real data are less meaningful than with simulated data. Moltz et al. [MBR*11] have shown in the case of manual liver tumor delineation that the expected error by using only one expert segmentation is between 17-35%. Table 2.10 summarizes the properties of data used for validation.

2.4. Conclusion

Several application domains exist, each with unique requirements. One methodology cannot fit all of them. The result of the vessel extraction methodology presented in this thesis, is the basis for interactive applications in the domain of interventional planning. Thus, an adequate hepatic vein extraction method fulfills the following requirements.

Short processing times. This follows from the declared goal to develop an interactive application, where relatively low response times are needed. An automated preprocessing of the data directly after acquisition is not always feasible, e.g. when the software is used by an external company that provides the resection planning as a paid to the clinic. Or when the software by itself is provided as a cloud service to the clinics.

Interactive. Typically, a complete vessel extraction methodology consists of several components linked together in a processing pipeline. Whenever an automatism fails in the pipeline, it should be possible to interactively correct the problem and then continue with the processing.

Intuitive. If user interaction is required, it should be done in an intuitive way.

When taking these requirements and the previous discussion into account, the following methods for the algorithmic processing pipeline illustrated in Figure 2.1 seem to be best suited (blue box in Figure 2.18).

- Vessel enhancement based on derivatives to enhance tubular structures in a preprocessing step.

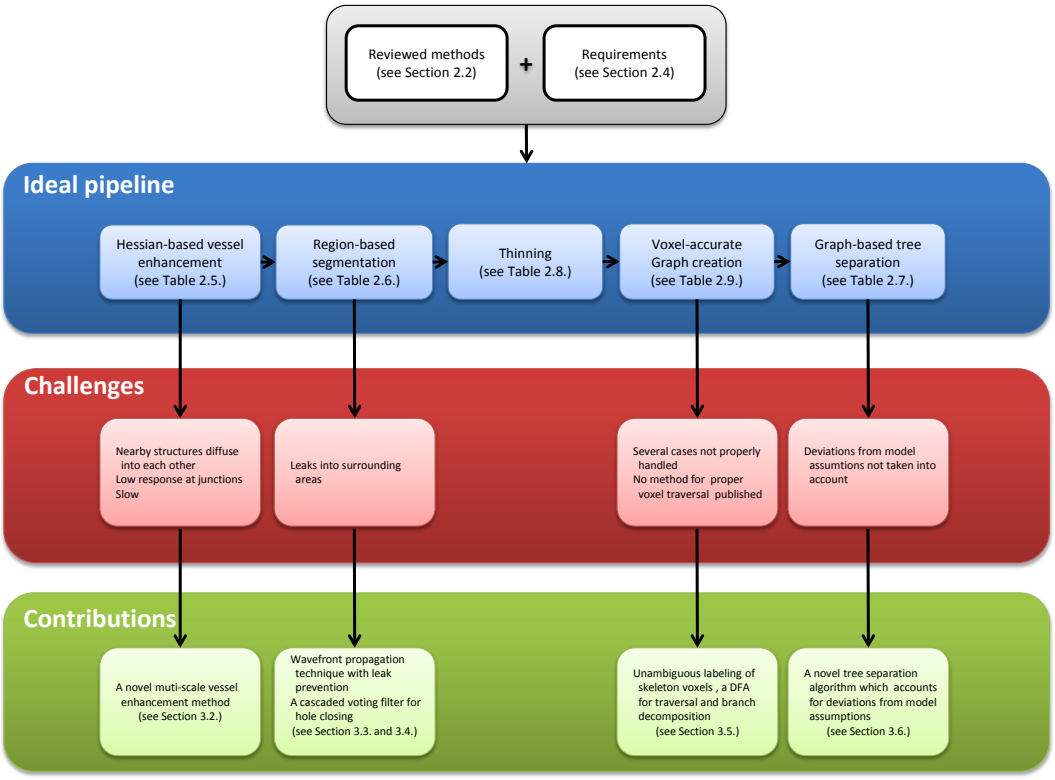


Figure 2.18.: When taking all the reviewed methods and requirements together with the discussion presented in Section 2.3 into account, an ideal pipeline for this work is derived (blue). However, existing methods have several shortcomings (red) which are addressed in this thesis (green).

- Region-based segmentation of the previously enhanced vessels.
- Voxel-accurate graph creation for further processing.
- Graph-based separation of interconnected hepatic veins.

However, current state of the art methods have several shortcomings (red box in Figure 2.18), which are discussed in the following together with proposed solutions (green box in Figure 2.18).

Vessel enhancement Derivative-based methods are embedded in a scale-space framework to detect tubular structures of different sizes. They utilize either the eigensystem of the Hessian, the gradient and its magnitude or a combination of both. The gradient captures the strength of an edge and the direction of the greatest rate of increase. In case of a bright tubular structure the direction is perpendicular to the center. The eigenvalues and eigenvectors of the Hessian provide local geometry information and are typically used to construct a vesselness measure by analyzing the local shape. Hereby, one vector points in the direction along the tubular structure, while the other two eigenvectors span a cross-sectional plane. The latter and the gradient vector were used to measure the circularity

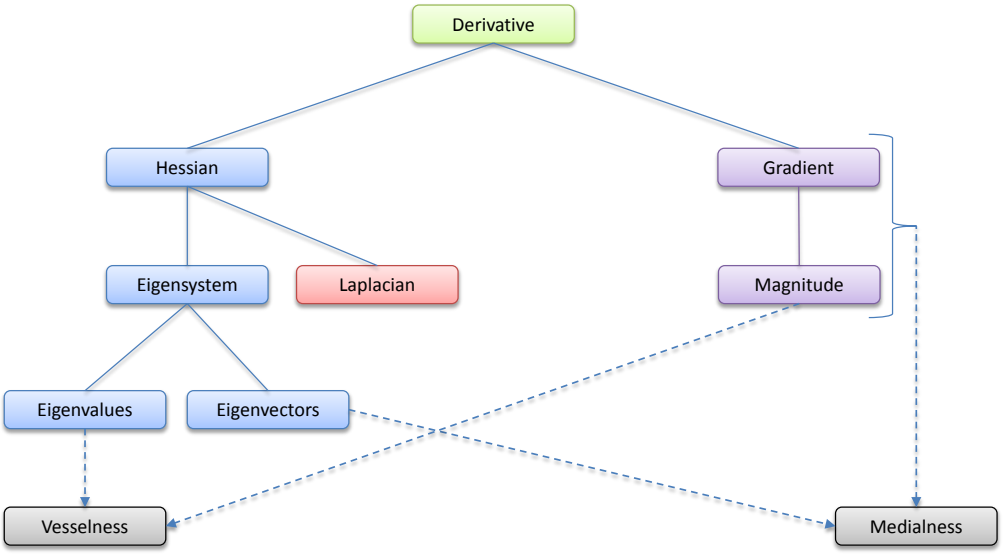


Figure 2.19.: Derivative-based vessel enhancement filters use the Eigensystem of the Hessian matrix (blue) or the gradient and its magnitude (purple) to implement a vesselness or medialness filter (gray). The Laplacian (red) was not used yet in any filter. It can be used to construct a vesselness function that performs equally well compared to other vesselness functions. Furthermore, it can be used to significantly reduce eigenvector and eigenvalue calculations.

at a given location by Krissian et al. [KMA*00] and Pock et al. [PBB05b]. Similar, the gradient magnitude has the highest response at edges and vanishes in the center of a bright tubular structure. This was used by Bennink et al. [BvAS*07]. Figure 2.19 illustrates the relationship between vesselness and medialness.

Hessian-based vessel enhancement methods are very popular, but suffer from the following problems.

- Nearby structures diffuse into each other
- Low filter response at junctions
- Many eigensystem calculations which affects performance

The vessel enhancement filter that is developed in this thesis is based on the following ideas. In a gradient vector field, the point in the center of a tubular structure can be considered as a source where all the gradient vectors originate. The idea is now to calculate a measure that quantifies how much a point behaves like a source for gradient vectors. This can be expressed as the divergence of the gradient at a given point, which results in the Laplacian of Gaussian when embedded in a scale-space framework. The Laplacian of Gaussian can be further used to significantly reduce the amount of eigensystem calculations for an image. Up to date, the Laplacian was not used for vessel enhancement (see Figure 2.19). It is part of this thesis and detailed in Section 3.2.4.

Most presented multi-scale vessel enhancement filters differ in the detection (vesselness/medialness) function and the underlying tubular model used to derive this function.

However, when it comes to combining different scales, all reviewed algorithms are identical. Namely, the maximum response among the scale space is used to create the final filter response. Thus, at a given position only information from one scale is used, although probably other scales could also contribute to the result. Section 3.2.3 deals with this problem and provides a solution. In summary, the following thesis statements are made.

- The Laplacian can be utilized to detect and enhance of tubular structures and significantly reduce calculations of eigenvalues and eigenvectors.
- All scales can contribute and improve the final filter response.

Segmentation Region growing algorithms are very efficient and easily initialized by placing a seed point within the object of interest. However, they have the following drawbacks.

- Leaks into neighboring areas render the result useless.
- Changing the threshold may reduce or prevent leaks at the cost of losing good parts of the segmentation.

A wavefront propagation technique was used and modified to detect and prevent leaks (e.g. [MSvdG*07]) in coronary arteries segmentation by imposing a threshold on the maximum allowed voxels that are added in every second iteration of the region growing algorithm. They used the fact that the diameters of the coronary arteries do not vary very much. This assumption does not hold in case of liver veins. Heimann et al. [HTKM04b] observed in the context of organ segmentation that object and leaked areas are usually characterized by a narrow bottelneck connection. This means that leaks usually start at small spots before they grow bigger. In Section 3.3, a Hessian-based vessel enhancement method is combined with a wavefront propagation technique to segment the hepatic veins. It prevents leaks by imposing a threshold on the minimum number of voxels added at each iteration.

Region-based methods tend to produce holes, bays and tunnels in the segmentation which may influence skeletonization and graph creation methods in a negative way. Especially when used in combination with vessel enhancement filters, this effect is pronounced at locations where the tubular model assumption is violated. The standard answer to this problem seems to be a morphological closing. However, morphological closing has the following disadvantages.

- Morphological closing creates connections between nearby vessels.
- Morphological closing thickens areas between vessels at bifurcations.

After segmentation, holes and tunnels are surrounded by ON-voxels. As discussed in Section 3.4, by taking the neighborhood of an OFF-voxel into account, it should be possible to fill holes without overly affecting the shapes of the vessels. This leads to the following thesis statements.

	case a	case b	case c	case d	Root needed
Gerig1993				-	no
Lee1994	-				no
Selle1999		-	-		no
Palagyi2006					yes
Chen2009			-	-	no

Table 2.11.: Comparison of voxel-accurate graph creation methods. Figure 2.11 shows the mentioned cases.

- Imposing a threshold on the minimum number of voxels added at each iteration prevents leaks and produces better results than a conventional region growing algorithm.
- A voting operator can fill holes, bays and tunnels and produces better results than morphological closing.

Graph-based tree separation The reviewed graph-based separation methods do not handle every situation equally well. The algorithm proposed by Selle et al. [SPSP02] includes only diameter information and assumes that connections appear where two trees clash together during traversal. The algorithm proposed by Homann et al. [HVN08] improves upon Selle's algorithm by including a-priori information about bifurcation angles. However, some preliminary experiments have shown that these model assumptions are often violated. For example, the diameter of a branch was not always decreasing and connecting edges not always the thinnest. This is usually due to noise and imperfect segmentations. Furthermore, it happened that obtuse angles at branches were present without being erroneous. Thus, the following thesis statement arises:

- Separation results can be improved, if violations to model assumptions, to a certain degree, are permitted.

This problem is addressed in Section 3.6.

Graph creation Graph-based separation methods require that a previously generated segmentation is transformed into a formal graph representation. All of the reviewed voxel-accurate graph creation methods fail to process some of the possible voxel combinations. Table 2.11 summarizes cases that are not properly handled by current state of the art methods.

Most of the methods try to find a solution by explicitly tagging one voxel as the main branch voxel. They will fail in case of symmetric situations. It seems that if a less restrictive policy is enforced then an algorithm can handle most (if not all) ambiguous situations. The following thesis statement is made and addressed in Section 3.5:

- Voxel-accurate graphs can be created, if voxels are properly classified and correctly traversed.

Validation No standardized evaluation methodology exists to date. Furthermore, only a few synthetic datasets are public available, but they have almost no practice-related value. Some clinical datasets are also available, but they come either with manual segmentations of unknown origin, or without them. In any case, manual segmentations are subject to intra- and interobserver variability and an objective quantitative evaluation is hard to perform. Simulated data might be a solution, but it was not used for quantitative evaluation of liver segmentation algorithms yet (see Table 2.4). This leads to the following thesis statement:

- A standardized evaluation framework based on simulated virtual data increases the value of quantitative evaluation in comparison to evaluation based on synthetic data.

Section 3.7 presents such a framework.

3. Hepatic vein extraction

Parts of this chapter were previously published by the author in the following conference papers:

- DRECHSLER K., OYARZUN LAURA C.: Comparison of vesselness functions for multiscale analysis of the liver vasculature. In *Proceedings of the 10th IEEE International Conference on Information Technology and Applications in Biomedicine (ITAB)* (2010), pp. 1–5
- DRECHSLER K., OYARZUN LAURA C.: A novel multiscale integration approach for vessel enhancement. In *Proceedings of the 23rd IEEE International Symposium on Computer-Based Medical Systems (CBMS)* (2010), pp. 92–97
- DRECHSLER K., OYARZUN LAURA C.: Hierarchical decomposition of vessel skeletons for graph creation and feature extraction. In *Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (2010), pp. 456–461
- DRECHSLER K., OELMANN S., OYARZUN LAURA C.: Separation of interconnected hepatic veins. In *Proceedings of the 24th IEEE International Symposium on Computer-Based Medical Systems (CBMS)* (2011), pp. 1–6
- DRECHSLER K., OYARZUN LAURA C., WESARG S.: Hepatic vein segmentation using wavefront propagation and multiscale vessel enhancement. In *Proceedings of SPIE Medical Imaging* (2013) (accepted)

3.1. Introduction

As explained in the introduction, a multiphase computed tomography (CT) scan of the abdomen is currently the best method to diagnose HCC [GKL*09]. Therefore, a contrast agent (COA) is injected intravenously, which enhances the hepatic vessels in several phases (see Section 3.7.3.1 for further explanation). Enhanced vessels in CT datasets are visually delineated from the surrounding parenchyma and suited as input to a segmentation algorithm.

In the following sections, a vessel extraction methodology is presented. It consists of several single processing steps, which can be easily replaced in case the requirements change. The focus was on usability and processing speed rather than full automatism and highest quality. The reason for this design choice was that the developed method built the basis for an interactive surgical planning application for open liver resections. This basically means that the main steps can be controlled by the user if the results are not satisfying. And because it is meant to be interactive, each processing step should be executed in acceptable time.

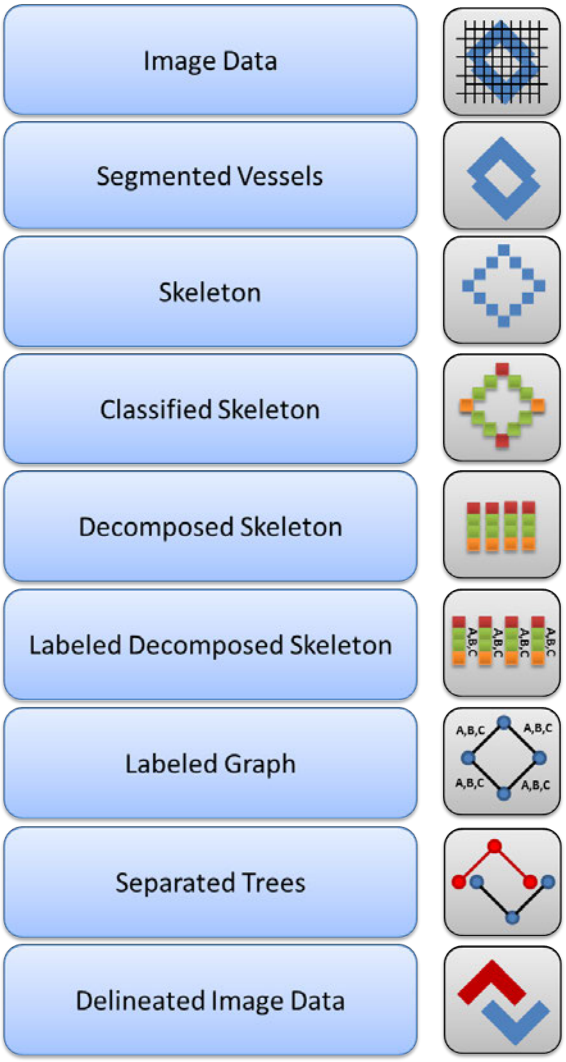


Figure 3.1.: Layer model consisting of several steps to extract hepatic veins from contrast enhanced CT data. The pictograms on the right side visualize the outcome of each layer. Image taken from [DOL10c].

Figure 3.1 illustrates the implemented dataflow model. It consists of seven abstractions from image acquisition to the final graph. At each layer, different algorithms can be chosen, depending on current state of the art, personal preference, targeted quantitative measures and so on. The output of one layer serves as input to the next layer. Image acquisition is the first step in this pipeline. Before further processing, the whole liver is segmented using a model-based approach proposed by Erdt et al. [EKSW10][1]. The reasons for this step are summarized as follows:

- It reduces the amount of data that will be processed in subsequent steps,
- it allows for an appealing visualization of the whole liver (Figure 3.2), and
- it simplifies vessel segmentation, because the liver vessels are connected to a huge

[1]The implementation used for this work was kindly provided by Marius Erdt.

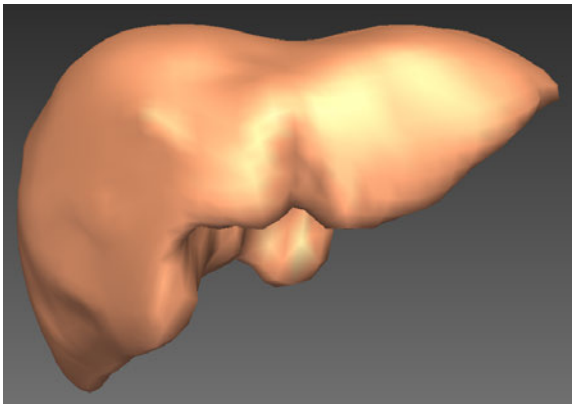


Figure 3.2.: Visualization of the liver segmented using the model-based approach by Erdt et al. [EKSW10].

vessel system spreading through the whole body, which are not present anymore after organ segmentation.

Afterwards, the vessels are segmented. The segmented vessels serve as input to a skeletonization algorithm that creates a one voxel wide, medial positioned and topology preserving representation of the vessel tree. The resulting skeleton serves as input to a voxel classifier that classifies each voxel into branch-voxel, regular-voxel and end-voxel. The output is then used in the next layer to decompose the skeleton into sub-branches. Each sub-branch is then stored voxel-wise in an easy to access data structure, which serves as input to the next layer which calculates attributes, like diameter, length and volume. Finally, this is used to create the graph representation. The graph serves as input to a tree reconstruction module that allows for computer assisted separation of interconnected trees. Finally, graph-based processing results are transferred back to the image domain.

The remainder of this chapter is organized as follows. In Section 3.2.1 the theory behind Hessian-based multiscale vessel enhancement is described. It lays the foundation for the following sections, where conceptually different vesselness functions are compared (Section 3.2.2), a multiscale integration approach using all scales is presented (Section 3.2.3) and novel line filter based on the Laplacian is presented (Section 3.2.4). In Section 3.5 the creation of formal graphs is described and in Section 3.6 a method for the separation of hepatic veins is proposed. Section 3.7 presents a vessel segmentation evaluation framework.

3.2. Vessel enhancement

3.2.1. Theory of Hessian-based multiscale analysis

It is known from differential geometry that the second order derivatives of an image can be geometrically interpreted by evaluating the eigenvalues at each point. The second order differential quantity for a volume $I(x, y, z)$ is given by the indefinite Hessian matrix

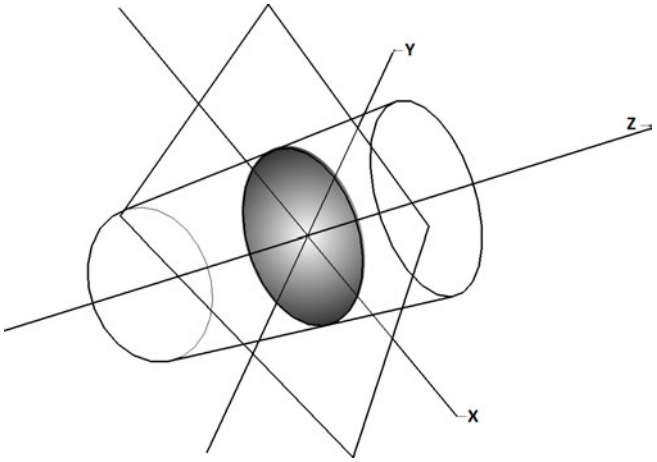


Figure 3.3.: A tubular object along the z-axis with a Gaussian profile. Image taken from [DOL10d].

$$H(x,y,z) = \begin{bmatrix} \frac{\partial^2 I}{\partial^2 x} & \frac{\partial^2 I}{\partial x \partial y} & \frac{\partial^2 I}{\partial x \partial z} \\ \frac{\partial^2 I}{\partial y \partial x} & \frac{\partial^2 I}{\partial^2 y} & \frac{\partial^2 I}{\partial y \partial z} \\ \frac{\partial^2 I}{\partial z \partial x} & \frac{\partial^2 I}{\partial z \partial y} & \frac{\partial^2 I}{\partial^2 z} \end{bmatrix}. \tag{3.1}$$

Let the eigenvalues of H be λ_1, λ_2 and λ_3 and their absolute values sorted in ascending order ($|\lambda_1| \leq |\lambda_2| \leq |\lambda_3|$). Furthermore, let v_1, v_2 and v_3 be the corresponding normalized eigenvectors. To explain the behavior of the eigensystem for tubular structures, an ideal model of a bright tube with a Gaussian profile (Figure 3.3) is assumed, which is formally given by

$$I(x,y,z) = e^{\frac{-(x^2+y^2)}{2\sigma^2}}. \tag{3.2}$$

Because the model is oriented along the z-axis the intensity values do not change along this direction and thus the second order derivatives are 0.

To explain the behavior of the eigensystem within the cross section of the tubular structure, either the x direction or the y direction can be examined. As can be seen in Figure 3.4, second order derivatives in directions perpendicular to the z-axis are negative in the interval $[-\sigma, +\sigma]$.

It can be shown that for the ideal tube $\lambda_1 = 0$, $\lambda_2 < 0$ and $\lambda_3 < 0$ [KMA*00]. In this case v_1 points along the tube while v_2 and v_3 span a cross section perpendicular to v_1. Knowledge about the correlation of eigenvalues and a specific shape can be used to construct a filter function to detect and enhance tubular structures in medical imaging modalities and to discriminate everything else.

The Hessian matrix contains information around a small local neighborhood of a point. Tubes with larger diameters are hardly detected. Therefore, scale space theory comes in handy to handle image structures at multiple scales. To get the scale space representation of an image it is convolved with a Gaussian function with different standard deviations for each scale. In the context of scale space theory the standard deviation is also known as the scale parameter. The normalized Gaussian function for a volume is given by

$$g_\sigma(x,y,z) = \frac{1}{(\sigma\sqrt{2\pi})^3} \cdot e^{\frac{-(x^2+y^2+z^2)}{2\sigma^2}}. \tag{3.3}$$

The volume at scale σ is then calculated by a Gaussian convolution as

$$L(x,y,z;\sigma) = g_\sigma(x,y,z) \star I(x,y,z). \tag{3.4}$$

Finally, Equation 3.1 and Equation 3.4 are combined to calculate the second order scale space of an image. Therefore, the Hessian matrix at scale σ is calculated as

$$H(x,y,z;\sigma) = \sigma^\lambda \begin{bmatrix} I \star \frac{\partial^2 g_\sigma}{\partial^2 x} & I \star \frac{\partial^2 g_\sigma}{\partial x \partial y} & I \star \frac{\partial^2 g_\sigma}{\partial x \partial z} \\ I \star \frac{\partial^2 g_\sigma}{\partial y \partial x} & I \star \frac{\partial^2 g_\sigma}{\partial^2 y} & I \star \frac{\partial^2 g_\sigma}{\partial y \partial z} \\ I \star \frac{\partial^2 g_\sigma}{\partial z \partial x} & I \star \frac{\partial^2 g_\sigma}{\partial z \partial y} & I \star \frac{\partial^2 g_\sigma}{\partial^2 z} \end{bmatrix}. \tag{3.5}$$

The factor σ^λ normalizes the intensity values across scale space and is needed to make responses at different scales comparable. The parameter λ was introduced by Lindeberg [Lin99] to define a family of normalized derivatives. After calculating the eigenvalues of H, the filter function is then applied to each scale σ in order to calculate the filter response $R(x,y,z;\sigma)$. Afterwards, the different scales are combined using the maximum response among the scale space to calculate the final response, denoted by

$$R(x,y,z) = \max_\sigma R(x,y,z;\sigma). \tag{3.6}$$

3.2.2. Comparison of three conceptually different vesselness functions

In the previous Section 3.2.1, the theory behind Hessian-based multiscale analysis was explained, but the description of a filter function $R(x,y,z;\sigma)$ was omitted. Many filter functions were proposed (Section 2.2.1.4), but not systematically evaluated. In this section, three conceptually different vesselness filter functions developed by Sato et al. (Section 3.2.2.1), Frangi et al. (Section 3.2.2.2) and Erdt et al. (Section 3.2.2.3) will be compared. Each author approached the problem differently, namely experimentally, geometrically and analytically. As a matter of fact, each method represents a class of possible solutions and was therefore chosen for this comparison. The chosen vesselness functions are explained in the following sections. A summary of their properties is presented in Table

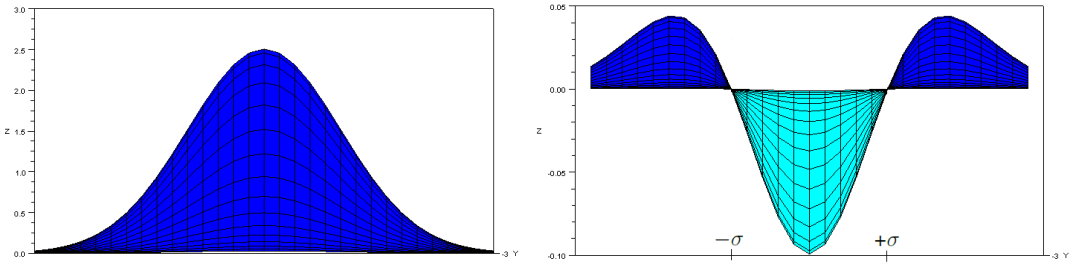


Figure 3.4.: Second order derivative of the Gaussian profile. Values in blue areas are positive while values in turquoise areas are negative. Image taken from [DOL10d].

	Sato	Frangi	Erdt
Approach	Experimental	Model-based	Analytical
EV Sort Order	$\lambda_n, \geq$	$\lvert\lambda_n\rvert, \leq$	$\lambda_n, \geq$
Used EV	2	3	3
Parameters	2	3	0

Table 3.1.: Properties of filter functions proposed by Sato et al., Frangi et al. and Erdt et al. Table taken from [DOL10a].

3.1.

3.2.2.1. Filter function by Sato et al.

Based on experimental analysis of an ideal tube model, Sato et al. [SNA*97] proposed a filter function given by

$$R(x,y,z;\sigma) = \begin{cases} e^{\frac{-\lambda_1^2}{2(\alpha_1\lambda_c)^2}} \cdot \lambda_c & \lambda_1 \leq 0, \lambda_c \neq 0 \\ e^{\frac{-\lambda_1^2}{2(\alpha_2\lambda_c)^2}} \cdot \lambda_c & \lambda_1 > 0, \lambda_c \neq 0 \\ 0 & \lambda_c = 0 \end{cases} \tag{3.7}$$

where $\alpha_1 < \alpha_2$ and $\lambda_c = min(-\lambda_2, -\lambda_3)$. As Sato et al. noticed, the exact shape of the function has no significant meaning, but it is important that it has an asymmetrical characteristic. Therefore, $\alpha_1 = 0.5$ and $\alpha_2 = 2.0$ were used. It is worth to mention that in their filter function only two eigenvalues are used and that the eigenvalues (not absolute eigenvalues) are sorted in descending order ($\lambda_1 \geq \lambda_2 \geq \lambda_3$).

3.2.2.2. Filter function by Frangi et al.

Frangi et al. [FNVV98] investigated the interpretation of the eigenvalues further and introduced a second order ellipsoidal model to distinguish plate-like, blob-like and tubular structures based on the eigenvalues of the Hessian matrix. In particular, they construct a dissimilarity measure that incorporates two geometric ratios to capture geometric information of the image. Furthermore they include the Frobenius matrix norm to reduce unpredictable filter responses for background voxels.

The first geometric ratio calculates the deviation from a blob-like structure, given by

$$R_B = \frac{|\lambda_1|}{\sqrt{|\lambda_2\lambda_3|}}. \tag{3.8}$$

The second ratio distinguishes between plate-like and line-like structures, given by

$$R_A = \frac{|\lambda_2|}{|\lambda_3|}. \tag{3.9}$$

The filter function is given by

$$R(x,y,z;\sigma) = \begin{cases} 0 & \lambda_2 > 0 \vee \lambda_3 > 0 \\ (1 - e^{\frac{R_A^2}{2a^2}}) e^{\frac{R_B^2}{2b^2}} (1 - e^{\frac{S^2}{2c^2}}) & \end{cases} \tag{3.10}$$

where S is the Frobenius norm of the Hessian matrix and a, b and c are parameters to control the sensitivity of the filter. The Frobenius norm is defined as

$$S = \|H\|_F = \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}^2 \tag{3.11}$$

with a_{ij} being the elements of the Hessian matrix. The parameters a and b were set to 0.5 (as used by Frangi et al. in [FNVV98]), while c was manually adjusted to get optimal results. Please note that Frangi et al. are using all three eigenvalues and that they sort them by sorting their absolute values in ascending order ($|\lambda_1| \leq |\lambda_2| \leq |\lambda_3|$).

3.2.2.3. Filter function by Erdt et al.

Erdt et al. [ERS08] approached the construction of an appropriate filter function completely analytically. The proposed filter function is given by

$$R(x,y,z;\sigma) = \kappa \cdot (\frac{2}{3}\lambda_1 - \lambda_2 - \lambda_3). \tag{3.12}$$

The factor κ is used to avoid the detection of very bright, plate-like structures and is given by

$$\kappa = 1 - \frac{||\lambda_2| - |\lambda_3||}{|\lambda_2| - |\lambda_3|}. \tag{3.13}$$

It is worth to note that the filter function by Erdt et al. is parameterless. Furthermore, their filter function uses all three eigenvalues which are sorted in descending order ($\lambda_1 \geq \lambda_2 \geq \lambda_3$).

3.2.2.4. Results

To evaluate the behavior of the filter functions, public available artificial datasets[2] were used. During all experiments the scale space was created using three scales with scale parameter $\sigma = 2, 3, 4$.

Synthetic data: Junctions The first experiment analyzes the behavior at junctions. Therefore, the top right three-dimensional image in Figure 3.5 was used. Gaussian noise with an increasing standard deviation from $\sigma = 0$ to $\sigma = 300$ was added. For a given position along the vertical profile shown as a red line in the top right image of Figure 3.5 (which goes through the center of the tubes) mean and standard deviation across all 300 images were measured. The result is the graph shown in the top left image of Figure 3.5. Afterwards, the filter functions proposed by Sato et al., Erdt et al. and Frangi et al.

[2] http://lmi.bwh.harvard.edu/research/vascular/SyntheticVessels/SyntheticVesselImages.html

with $c = 5$ and $c = 32$ were applied and the described calculations were repeated. The corresponding graphs are shown in the remaining rows of Figure 3.5 in the given order. It can be seen that the filter functions by Sato et al. and Erdt et al. produced very similar results. Furthermore, it can be seen that the filter function proposed by Frangi et al. has problems in the area of the junction. While for $c = 32$ this is obvious just by looking at the graph (the green and red lines have a contact point), the graph for $c = 5$ shows no problems, because the measured vertical profile is detected very well. However, by looking at the produced images, it can be seen that the area at the junction around the measured vertical profile is very weak. Other values for c show similar results. Simply spoken, the filter function proposed by Frangi et al. tends to disconnect junctions. The images next to each graph confirm these observations for the case where no noise was added to the original image.

Synthetic data: Nearby vessels The next experiment analyzes the behavior for nearby vessels. Therefore, the top right three-dimensional image in Figure 3.6 was used. Like in the first experiment, Gaussian noise with an increasing standard deviation from $\sigma = 0$ to $\sigma = 300$ was added for a given position along the vertical profile shown as a red line in the top right image of Figure 3.6 and mean and standard deviation across all 300 images were measured. The result is the graph shown in the top left image of Figure 3.6. The other rows show the results after applying the filter functions proposed by Sato et al., Erdt et al. and Frangi et al with $c = 5$ and $c = 32$. Again, the results produced by Sato et al. and Erdt et al. are very similar. However, this time the filter function developed by Frangi et al. produces much better results than the other two filter functions. Both tubes are clearly separated in the graph. This has an important implication for a consecutive segmentation step. In this case, it is much easier to segment nearby vessels detected by the filter function proposed by Frangi et al..

Real data Next, the filter functions were applied to real contrast enhanced datasets of the liver. Figure 3.7 shows a representative detail of one of these datasets. It confirms what was found in the first experiment, namely that the filter function proposed by Frangi et al. tends to disconnect junctions. This behavior can also be seen in results published by different authors (e.g. [QBD*09]). The parameter c efficiently controls how much noise is removed, which makes a successive segmentation step easier. The filter functions proposed by Sato et al. and Erdt et al. produced very similar results and, in comparison to the filter function by Frangi et al. did not disconnect branches.

The vessels of the liver were segmented using three dimensional region growing with a manually selected seed point at the root of the portal vein. The thresholds were chosen manually and, if necessary, iteratively adapted until the results were satisfactory. One of the results is shown in Figure 3.8. There is no significant difference when region growing is applied to the results of the filter functions proposed by Sato et al. and Erdt et al.. There is also no significant difference when region growing is applied to the output of the filter function proposed by Frangi et al. with $c = 5$ and $c = 32$. However, it was much harder to find an appropriate threshold for the former output, because of oversegmentations due to leakings into noisy areas.

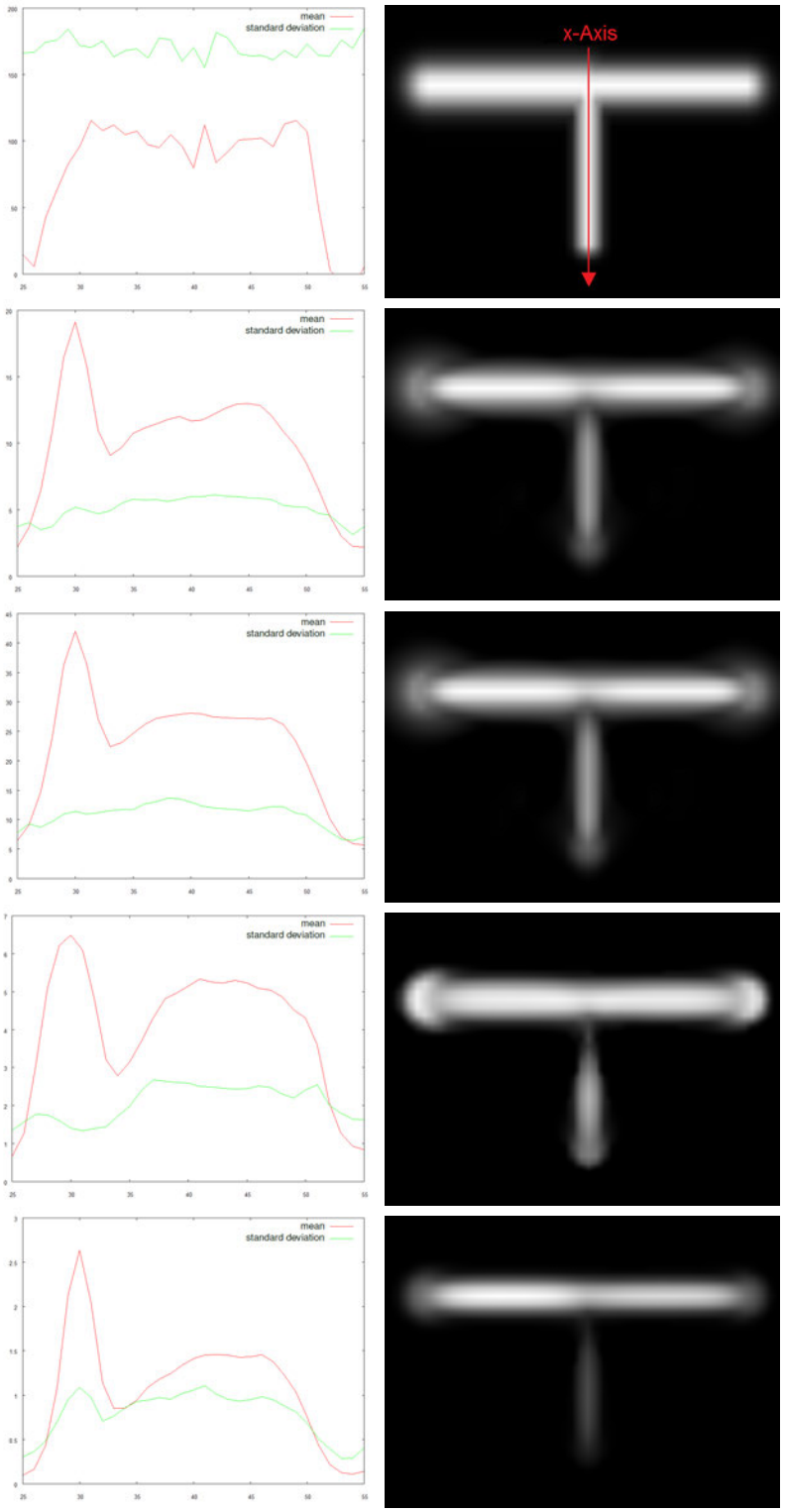


Figure 3.5.: Experimental results using synthetic junctions. The first row shows the mean (red) and standard deviation (green) of the vertical profile (red line in the top right image) measured using 300 images with increasing Gaussian noise (x-axis: position, y-axis: mean/standard-deviation). The remaining rows show the results after multiscale analysis using filter functions by Sato et al., Erdt et al., Frangi et al. with c = 5 and Frangi et al. with c = 32. The images in the right column were produced without added noise. Image taken from [DOL10a].

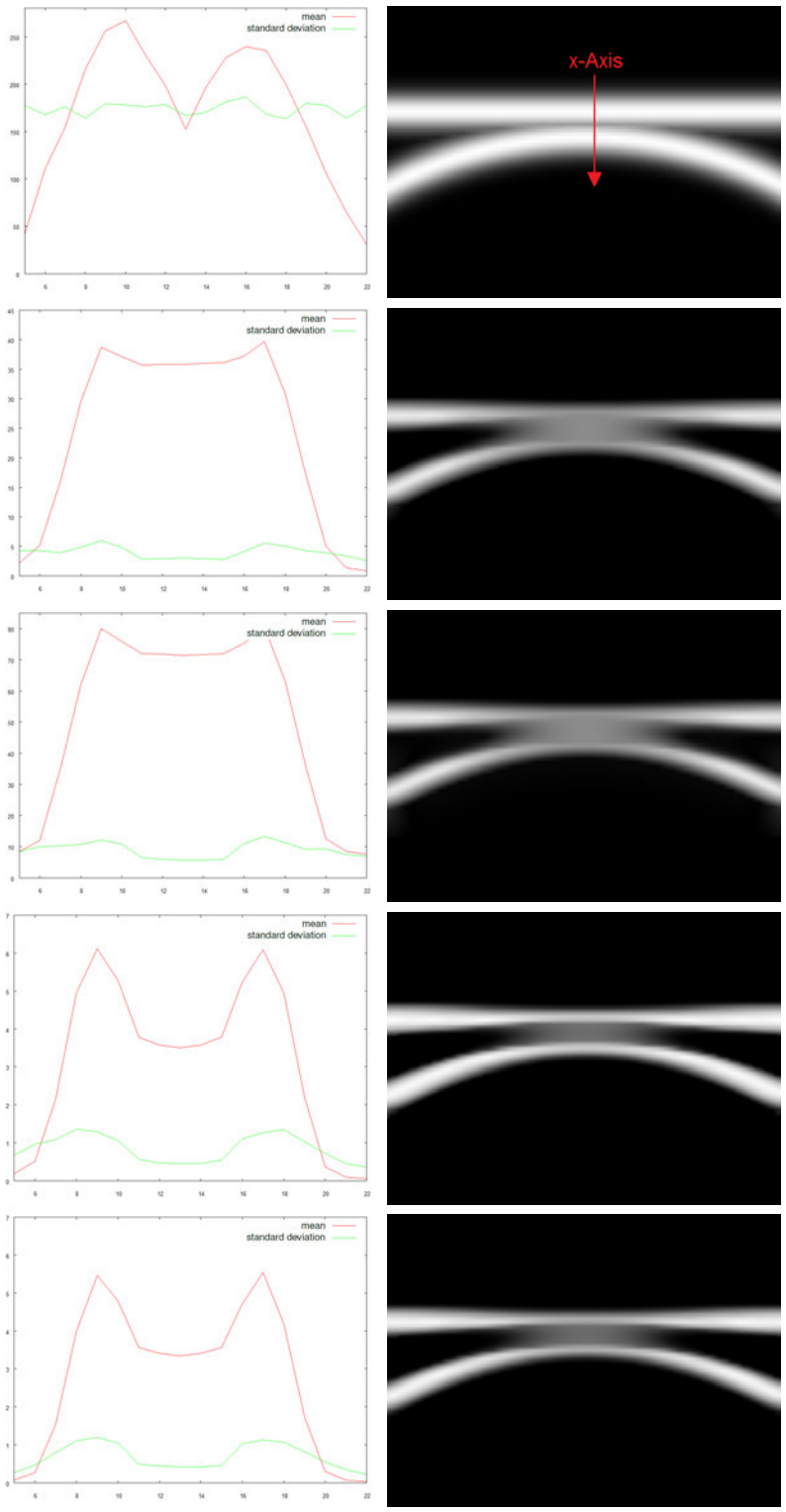
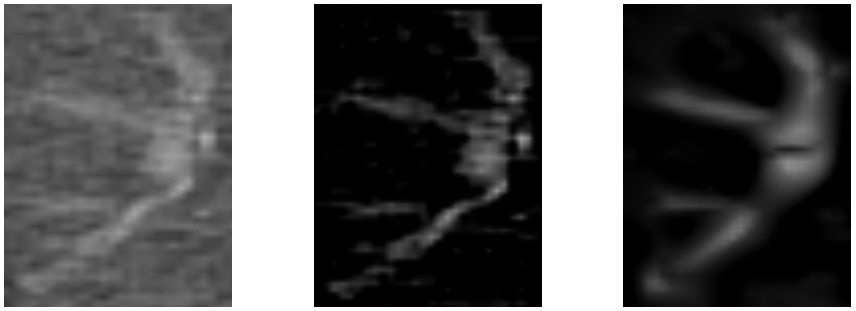


Figure 3.6.: Experimental results using synthetic nearby vessels. The first row shows the mean (red) and standard deviation (green) of the vertical profile (red line in the top right image) measured using 300 images with increasing Gaussian noise (x-axis: position, y-axis: mean/standard-deviation). The remaining rows show the results after multiscale analysis using filter functions by Sato et al., Erdt et al., Frangi et al. with $c = 5$ and Frangi et al. with $c = 32$. The images in the right column were produced without added noise. Image taken from [DOL10a].



(a) Original (b) Adjusted (c) Sato et al.

(d) Erdt et al. (e) Frangi et al.,c = 5 (f) Frangi et al.,c = 32

Figure 3.7.: Comparison using a CT dataset of the liver. (a) Representative detail of the liver from a CT dataset. (b) The same detail with a different window leveling. Results using filter functions by (c) Sato et al., (d) Erdt et al., (e) Frangi et al. with c = 5 and (f) Frangi et al. with c = 32. Image taken from [DOL10a].

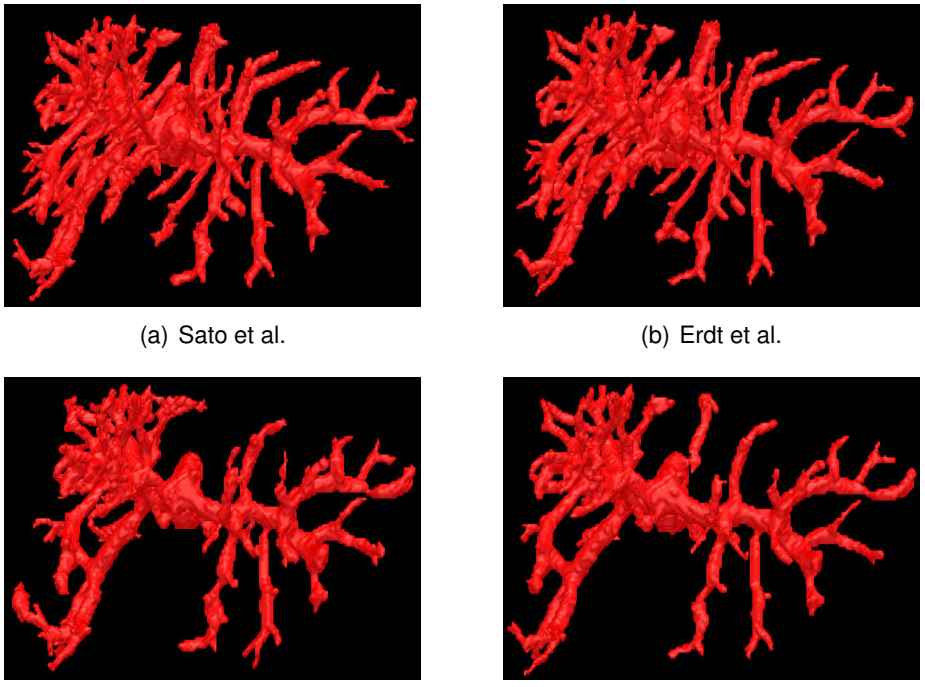
	Sato et al.	Erdt et al.	Frangi et al.
Junctions coherence	+	+	-
Nearby vessels separation	-	-	+
Separation from background	-	-	+
Function parameters	+	+	-

Table 3.2.: Rating of multiscale vessel enhancement methods proposed by Sato et al., Frangi et al. and Erdt et al. based on described evaluation results.

3.2.2.5. Discussion and conclusion

Three conceptually differently developed vesselness measures proposed by Sato et al., Frangi et al. and Erdt et al. were compared with segmentation of the liver vasculature in mind. Surprisingly, the functions proposed by Sato et al. and Erdt et al. behaved very similar.

The strength of both is the coherence of the output at junctions, whereas nearby vessels are hardly separated and tend to diffuse into one another. Furthermore, both filter functions are parameterless (Erdt et al.) or quasi parameterless (Sato et al.) because the parameters can be kept constant. The function proposed by Frangi et al. has its strength in the suppression of background noise and separation of nearby vessels, whereas vessels tend to get disconnected at junctions. However, parameters must be manually adjusted to get optimal results. Table 3.2 summarizes these findings.



(a) Sato et al. (b) Erdt et al.

(c) Frangi et al. with $c = 5$ (d) Frangi et al. with $c = 32$

Figure 3.8.: Region growing applied to the results of different filter functions by (a) Sato et al., (b) Erdt et al., (c) Frangi et al. with $c = 5$ and (d) Frangi et al. with $c = 32$. Image taken from [DOL10a].

The vessel system of the liver is very dense and due to partial volume effects and motion artifacts it happens that nearby portal vein and liver vein branches diffuse into one another in the acquired CT image. These cases are better handled by the filter function proposed by Frangi et al.. However, in experiments with real CT datasets it was observed that there are only a handful of connected areas in a dataset and efficient algorithms were proposed to algorithmically separate connected vessels of two different vessel systems of the liver [SPSP02, HVN08]. For this reason, it is more advantageous to have properly connected branches, which is achieved with either the filter function proposed by Sato et al. or Erdt et al..

3.2.3. Multiscale integration using all scales

As explained in Section 3.2.1, a vesselness measure is calculated for each scale and then the maximum response across all scales is used to produce the final response. It was shown in Section 3.2.2, that the vesselness functions proposed by Sato et al. and Erdt et al. are suboptimal at locations with nearby vessels. The maximum response approach used for multiscale integration introduces problems due to imperfect vesselness measures and Gaussian blurring. The former leads to disturbances in the final response caused by false positives across the scale space. The latter causes nearby vessels to diffuse into one another and are consequently detected as one. Subsequent processing steps (e.g. segmentation) suffer from these problems. An ideal filter that perfectly distinguishes vessels from non-vessels would not cause these problems. That is why different

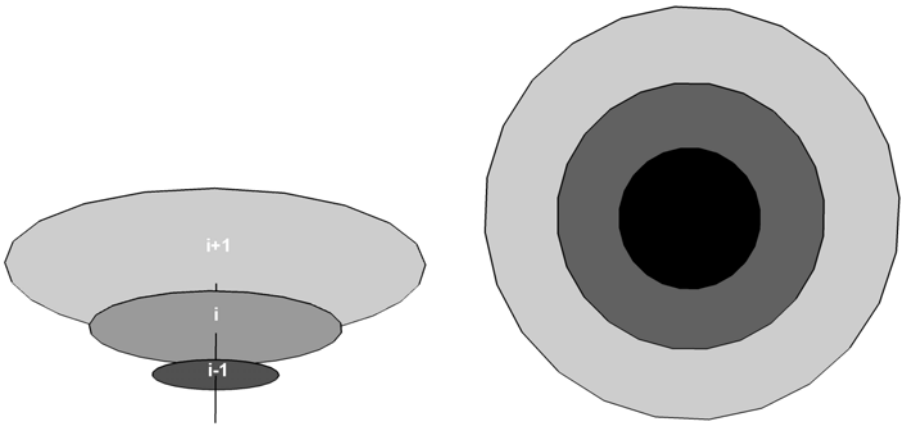


Figure 3.9.: (a) Vesselness filtered scale space representation of a tubular structure. (b) Result after adding the vesselness measures of each scale. Darker colors represent higher values. Image taken from [DOL10d].

vesselness measures were proposed that try to suppress non-vessel areas as much as possible. The previous chapter compared three of them. However, in practice they still produce false positive responses in areas where no vessels are present, which leads to the described problems.

In the following, a multiscale integration approach called *Weighted Additive Response* that overcomes the described limitations is proposed. It makes use of the fact that every scale can add to the result, not only the scale where the response is maximal.

So far, most publications differ in the vesselness function and the underlying tubular model used to derive this function. However, when it comes to combining different scales, the majority (if not all) of published papers (e.g. [SNA*97, FNVV98, KMA*00, PBB05b, BvAS*07, ERS08]) are identical. Namely, the maximum response among the scale space is used to create the final response, denoted by

$$R(x, y, z) = \max_{\sigma} R(x, y, z; \sigma). \tag{3.14}$$

This approach works quite well if the vesselness function is very reliable, such that noise around the tubular structures to be segmented is efficiently removed. However, in experiments with contrast enhanced liver datasets, the vesselness function (Equation 3.7) applied at different scales produced false positive responses. If Equation 3.14 is then used to produce the final response, all these false positives negatively influence the result. Even worse, the maximum false positive response across all scales is used by Equation 3.14 for a given location. This leads to a final response where vessels (true positive responses) can be hardly distinguished from surrounding false positives. This, in turn, makes it harder to segment the vessels and visualizing the vessel enhanced volume with volume rendering techniques shows a lot of noise (see Figure 3.54 on page 105).

The main idea behind the proposed approach is that each scale can contribute to the result. To explain this, imagine that the vesselness is calculated at scales $m, ..., n$, $m < n$. Let the maximum response of a tube be at scale i, $m < i < n$. As illustrated in the

right image of Figure 3.9, the diameters of the responses at scales below *i* are focused around the center of the tube which is decreasing with increasing distance to scale *i*. The opposite behavior can be noted for scales above *i*. Here the diameter of the response is increasing with increasing distance from scale *i*. This is due to Gaussian blurring with convolution kernels bigger than the vessel diameter.

Figure 3.10 shows scale space responses for tubular structures with Gaussian and bar-like profiles using Sato's vesselness function. Although the filter responses are differing for a given diameter, it can be seen that at each scale a response is produced that could contribute to the final filter response. The maximum response integration does not make use of this.

This observation can be used to focus the enhancement of a vessel around its center. Therefore, the final response is produced by adding the weighted contributions of each scale (Figure 3.9(b)). This is expressed by the following generalized equation which is called the *Weighted Additive Response*

$$R(x, y, z) = \sum_{\sigma} w_{\sigma} \cdot R(x, y, z; \sigma). \tag{3.15}$$

If $w_{\sigma} = \frac{1}{\sigma^{\lambda}}$ is chosen, then the effect of scale space normalization (factor σ^{λ} in Equation 3.5) is reversed. A similar effect can be reached by choosing $w_{\sigma} = 1$ and by removing the factor σ^{λ} in Equation 3.5. In this case, the filter responses correlate with the amplitude of the normalized Gaussians, which decrease with increasing scale parameter. Thus, Equation 3.15 has implicit 'natural' weights giving lower scales a higher importance than higher scales.

If $w_{\sigma} = 1$ is chosen (without eliminating σ^{λ} in Equation 3.5), then the final response is just the added responses across the normalized scale space.

It could make sense to use different shaped weight factors across scale space to take into account that low and high scales are too noisy or too blurred to be included in the final response.

Furthermore, with $w_{\sigma} = -1$ the result is similar to image enhancement using Differences of Gaussians (DoG).

3.2.4. A Laplacian-based vesselness filter

As explained in section 2.2.1.4, derivative-based methods utilize either the eigensystem of the Hessian, the gradient and its magnitude or a combination of both.

The gradient magnitude is quite effective at detecting sharp edges where intensity values change abruptly, which usually happens at vessel borders. It was used by different authors to construct a medialness measure, which samples the gradient magnitude along a circular shape, which is oriented using the eigenvector from the Hessian that points in the tangent direction when located at the center of a vessel.

The gradient was used in a similar way by utilizing the fact that the gradient vectors point away from the center of a vessel and are perpendicular to the vessel's tangent direction. A medialness measure was used by verifying that the eigenvectors perpendicular to the tangent direction point in the same direction as the gradient vectors at the vessel borders.

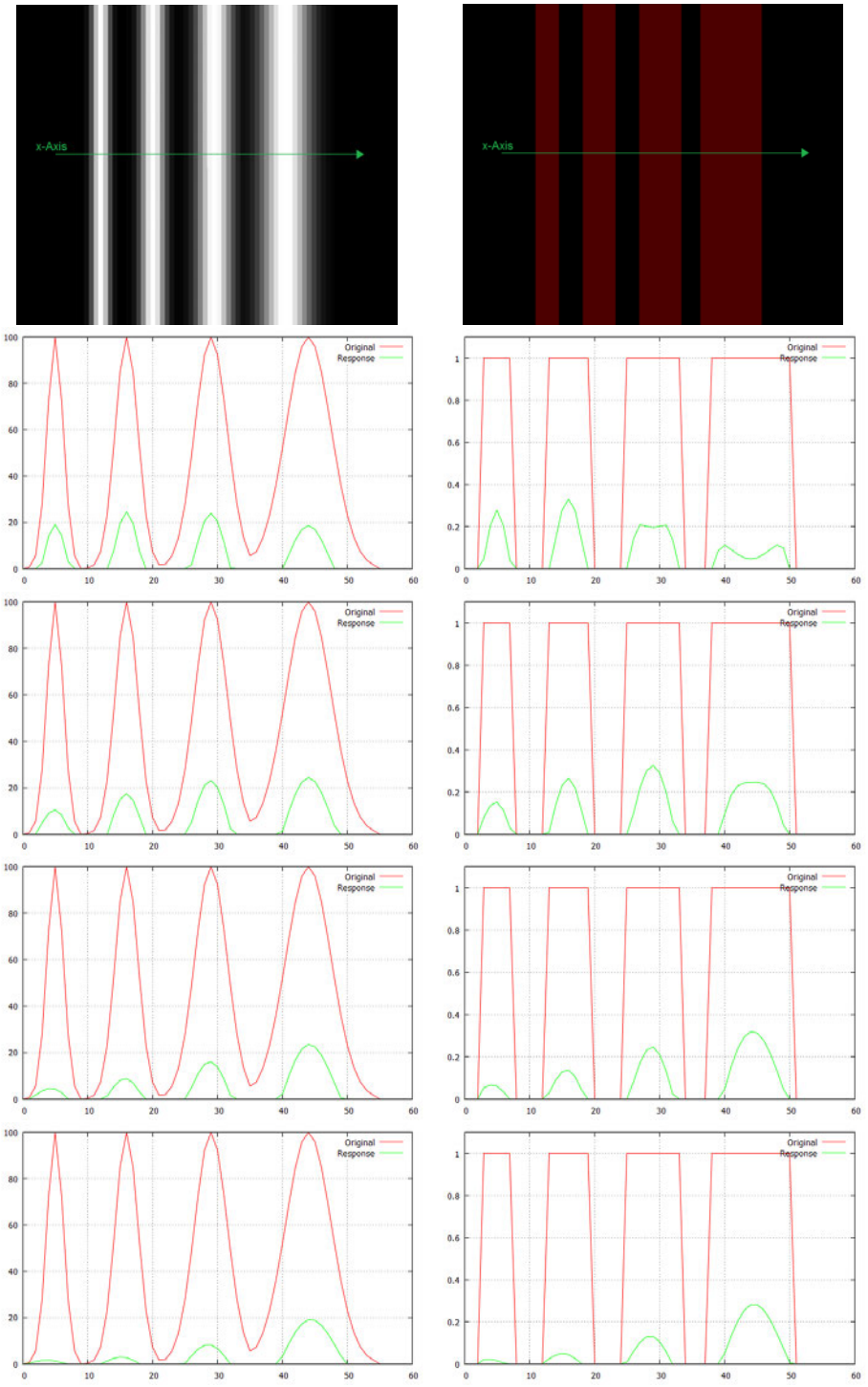
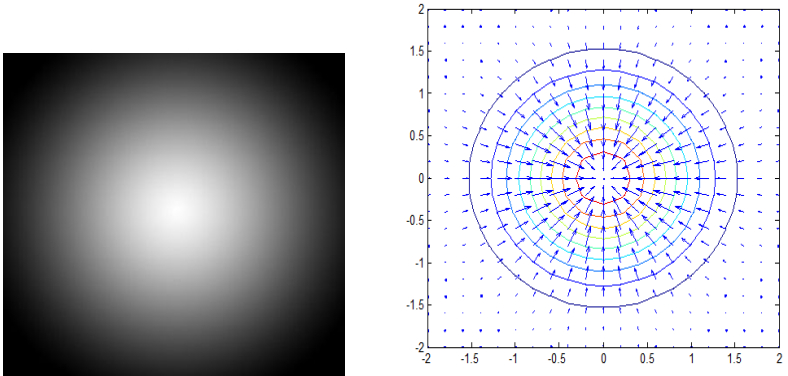


Figure 3.10.: Scale-space responses for Gaussian and bar-like profiles using Sato et al.'s. vesselness function. Top row: Original images. Second to fifth row: Vesselness responses (green) at scale $\sigma = 2.0, 3.0, 4.0$ and 5.0 respectively for the corresponding profile (red).



(a) (b)

Figure 3.11.: (a) Cross section of a tubular structure with Gaussian profile. (b) Gradient vector field for the Gaussian profile. It can be seen that the center of the tubular structure acts as a source or sink (depending on the sign) for the gradient vectors.

As just explained, the gradient vectors for a tubular structure with Gaussian profile point away from the vessel center in direction of the vessel borders (Figure 3.11). The point in the vessel center can therefore be considered as a source where all the gradient vectors originate. The idea is now to calculate a measure that quantifies how much a point behaves like a source. This is called divergence. The divergence of the gradient results in the Laplacian (Section 3.2.4.1) and when embedded in a scale-space framework, in the Laplacian of Gaussian (Section 3.2.4.2). The Laplacian of Gaussian is a second-order differential operator that will be used as an alternative way to detect tubular structures (Section 3.2.4.5). It will be shown that the Laplacian of Gaussian is equivalent to the core of the analytically derived filter function proposed by Erdt et al. [ERS08] (Section 3.2.4.3). Furthermore, the Laplacian can be used to significantly speed-up eigenvalue calculations for a given region (Section 3.2.4.4).

3.2.4.1. Divergence of the gradient: The Laplacian

The divergence of a vector field F measures the extent to which a point p behaves as a source or a sink. Intuitively, the divergence is the sum of all sources minus the sum of all sinks. Formally, it is defined as the limit of the net flow of F across the closed surface S of a three dimensional region R divided by the volume V of that region as it converges to 0. Formally,

$$div\ F(p) = \lim_{V \to 0} \int_S \frac{F \cdot n}{V} dS \tag{3.16}$$

where n is the unit surface normal to S at each point. It can be shown that the divergence of a continuously differentiable vector field F in Cartesian coordinates is given by

$$div\ F = \nabla \cdot F = \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} + \frac{\partial F_z}{\partial z}. \tag{3.17}$$

When applied to the gradient field of a scalar valued image, it becomes a second-order differential operator given by

$$div\ (grad\ I) = \nabla \cdot (\nabla I) = \nabla^2 I = \Delta I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} + \frac{\partial^2 I}{\partial z^2}. \tag{3.18}$$

The divergence of the gradient is also known as the Laplacian (or Laplace operator). It can be directly extracted from the Hessian matrix (Equation 3.1) by adding all diagonal elements.

3.2.4.2. Scale-space divergence of the gradient: The Laplacian of Gaussian

A second-order differential operator like the Laplacian is known to be highly sensitive to noise. Thus, an image is usually smoothed with a Gaussian filter before calculating the Laplacian. The resulting second-order differential operator is then called the Laplacian of Gaussian. Formally,

$$div\ (grad\ (g_\sigma * I)) = I \star \frac{\partial^2 g_\sigma}{\partial x^2} + I \star \frac{\partial^2 g_\sigma}{\partial y^2} + I \star \frac{\partial^2 g_\sigma}{\partial z^2} \tag{3.19}$$

where $\star$ is the convolution operator. The Laplacian of Gaussian can be instantly obtained from the scale-space Hessian matrix (Equation 3.5), which is calculated for different scales by prior Gaussian blurring, by adding all diagonal elements.

3.2.4.3. Similarity to Erdt et al.'s filter function

As described in Section 3.2.2.3, Erdt et al. derived a solution for an optimal filter (with respect to a tubular model) as

$$f = \underbrace{\kappa}_{\text{Isotropy factor}} \cdot \underbrace{(\frac{2}{3}\lambda_1 - \lambda_2 - \lambda_3)}_{\text{Analytically derived part}}. \tag{3.20}$$

The isotropy factor κ (see Equation 3.13) calculates the deviation from a plate-like structure, while the second part was analytically derived. It will be shown now that the analytically derived part (in the following called Erdt's equation) equals the Laplacian in the center of a tubular structure.

Lemma 3.1. Let H be the Hessian matrix of an Image *I* and $\lambda_1, \lambda_2, \lambda_3$ its corresponding eigenvalues sorted in ascending order ($\lambda_1 \geq \lambda_2 \geq \lambda_3$). Then

$$\frac{2}{3}\lambda_1 - \lambda_2 - \lambda_3 = -\Delta I \tag{3.21}$$

in the center of a tubular structure.

To proof this lemma, the following lemma is needed.

Lemma 3.2. If *A* is a square matrix with dimensions *n* with real entries and if $\lambda_1, \ldots, \lambda_n$ are the eigenvalues of *A* then the trace *Tr* of matrix A equals the sum of the eigenvalues

of A:

$$Tr(A) = \sum_i a_{ii} = \sum_i \lambda_i. \tag{3.22}$$

Proof of Lemma 3.1. Starting at the left-hand side of Equation 3.21 it follows that

$$\begin{aligned}
\frac{2}{3}\lambda_1 - \lambda_2 - \lambda_3 &= -(\lambda_1 + \lambda_2 + \lambda_3) + \frac{5}{3}\lambda_1 && \text{(Rearranging Equation 3.21)} && (3.23)\\
&= -\sum_{i=1}^{3} \lambda_i + \frac{5}{3}\lambda_1 && && (3.24)\\
&= -Tr(H) + \frac{5}{3}\lambda_1 && \text{(Using Lemma 3.2)} && (3.25)\\
&= -\sum_{i=1}^{3} a_{ii} + \frac{5}{3}\lambda_1 && \text{(Definition of Trace)} && (3.26)\\
&= -(\frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} + \frac{\partial^2 I}{\partial z^2}) + \frac{5}{3}\lambda_1 && \text{(Elements of Hessian)} && (3.27)\\
&= -\Delta I + \frac{5}{3}\lambda_1 && \text{(Using Equation 3.18).} && (3.28)
\end{aligned}$$

As can be seen in the last line (3.28), the analytically derived part of Equation 3.20 adds the term $\frac{5}{3}\lambda_1$ to the Laplacian. In the center of a vessel $\lambda_1 = 0$ and thus Equation 3.21 holds.

It was shown now that the Laplacian and Erdt's equation are equal in the center of a tubular structure. Two other positions can be distinguished. The filter can be

1. outside a vessel-like structure or
2. inside a vessel-like structure, but not in the center.

In the first case, it is not predictable which values the eigenvalues will have. Thus, the Laplacian and Erdt's equation produce equally good or equally bad results, with the difference that the results differ by $\frac{5}{3}\lambda_1$.

In the second case, λ_1 will still point towards the tangent direction of an ideal tubular model. Only the relation between λ_2 and λ_3 will change. Hence, the Laplacian and Erdt's equation will return equal results. □

3.2.4.4. Runtime optimization using the Laplacian

An algorithm which needs the eigenvalues of the Hessian to compute a vesselness measure has to solve the eigenvalue problem for each voxel of the desired region. However, the calculation of eigenvalues is the most expensive operation in these algorithms. Therefore, it is desirable to decide in an earlier stage for each voxel, if the calculation of the eigenvalues is necessary. This is the case when a voxel belongs to a vessel. The Laplacian can be used to reduce unnecessary eigenvalue computations. As already explained in the previous sections, the Laplacian is equivalent to calculating the trace of the Hessian, which is equivalent to the sum of all eigenvalues (see Equation 3.22). Furthermore, inside a bright vessel $\lambda_1 \approx 0$ and λ_2 and $\lambda_3 << 0$. Thus, the sum of these eigenvalues is

always negative. Differently speaking, if the sum of these eigenvalues is positive, further computations for a particular voxel can be skipped. In Section 3.8.2, the speed-up using this approach is evaluated.

3.2.4.5. Proposed filter function

Until now it has been argued that the Laplacian can be used to detect voxels belonging to a tubular structure and to speed-up further processing by looking at its sign. However, a negative sign is not enough to detect tubular structures only. This is because gradient vectors also originate from the center of blobs and plate-like structures. Thus, the sum of eigenvalues will be negative for these objects, too. To solve for this, the isotropy factor proposed by Erdt et al. [ERS08] is used, which is given by

$$\kappa = 1 - \frac{||\lambda_2| - |\lambda_3||}{|\lambda_2| + |\lambda_3|}. \tag{3.29}$$

The isotropy factor κ approaches 1, if $|\lambda_2|$ and $|\lambda_3|$ have approximately the same value. This is the case in the center of a tubular structure, while at other locations it approaches zero. The final filter function is given by

$$f = \begin{cases} g, & \text{if } \Delta(g_\sigma \star I(x)) < 0 \\ 0, & \text{else} \end{cases}$$

$$g = \begin{cases} -\kappa \cdot \Delta(g_\sigma \star I(x)), & \text{if } \lambda_2 < 0 \text{ and } \lambda_3 < 0 \\ 0, & \text{else.} \end{cases}$$

The vesselness function f is applied to each scale and the result is integrated using Equation 3.15.

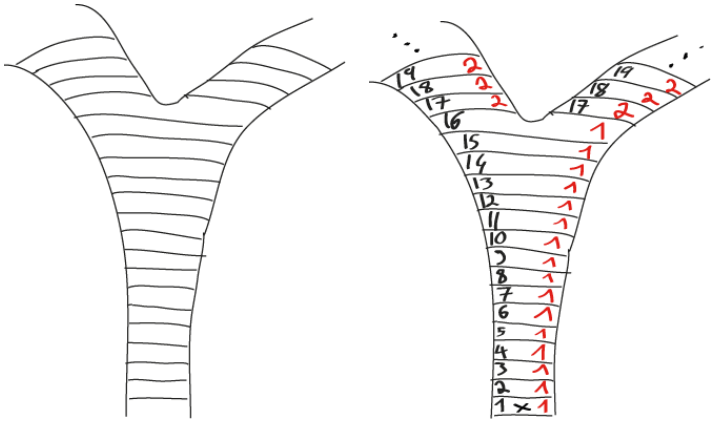
3.3. Segmentation using wavefront propagation

Wavefront propagation can be regarded as ordered region growing scheme that evolves an interface inside a vessel. Formally, it can be defined as follows.

Let O_0 be a set of user provided seed points inside an object and $N_\alpha(x)$ a set of voxels connected to x in a α-neighborhood with $\alpha = 6, 18, 26$ in 3D. Furthermore, let $H(x)$ be a similarity criterion. The algorithm can then be recursively defined as

$$O_{n+1} = \{x | \exists o \in O_n . x \in N_\alpha(o) \wedge H(x)\}. \tag{3.30}$$

Each recursion step adds another layer (or wave) to the segmentation (Figure 3.12(a)) which propagates the front inside the vessel. Voxels within a layer are uniquely numbered (n) and counted ($|O_{n+1}|$). This is utilized to detect and prevent leaks. The algorithm proposed by Metz et al. [MSvdG*07], for example, was developed to segment the coronary arteries. It detects if too many voxels are added with each new layer. They assume that if too many voxels are added it is a sign for a leak. Unfortunately, the liver vessels vary in diameter much more than the coronary arteries. Thus, their assumption is not applicable



(a) (b)

Figure 3.12.: (a) Wavefront propagation techniques add in each iteration a new layer to the segmentation. (b) Each voxel in a layer can be consecutively numbered. This is utilized to detect leaks. Furthermore, branches can be detected using a connected component analysis. This can be utilized to 'grow back' a whole branch if a leak is detected.

in this case. Furthermore, the previously described vessel enhancement filter suppresses most of the areas that have not a tubular nature. As a matter of fact, leaks usually appear at narrow bottlenecks to neighboring artifacts that have a tubular shape. To prevent this kind of leaks, Equation 3.30 is enhanced as follows.

$$O_{n+1} = \begin{cases} \{x | \exists o \in O_n . x \in N_\alpha(o) \wedge H(x)\} & \text{if } |O_n| >= T_{leak} \\ \{\emptyset\} & else, \end{cases} \tag{3.31}$$

where T_{leak} is a threshold on the minimum number of voxels of the current layer necessary to continue growing the branch.

The connectivity of the propagating interface can be analyzed to detect bifurcations and grow each branch independently. Hereby each disconnected component represents the start of a new branch (Figure 3.12(b)). This can be utilized in two scenarios. First, if a leak occurs, the last branch that led to the leak can be completely 'ungrown'. Second, if only one vessel system is visible, the Strahler scheme can be used to number each branch. This allows for the removal of outer branches such that only main/important branches are visible. Hahn et al. [HPSP01] used it in their system for the exploration of vessel structures to assign the same value to vessels with approximately the same diameter and volume. The Strahler number for each branch is assigned as follows.

- If a branch has no sub-branches, its Strahler number is one.
- if a branch has one sub-branch with Strahler number n, and all other sub-branches have Strahler number less than n, then the Strahler number of that branch is n.
- If a branch has one or more sub-branches with Strahler number n, and no sub-branch has a greater Strahler number, then the Strahler number of that branch is $n+1$.

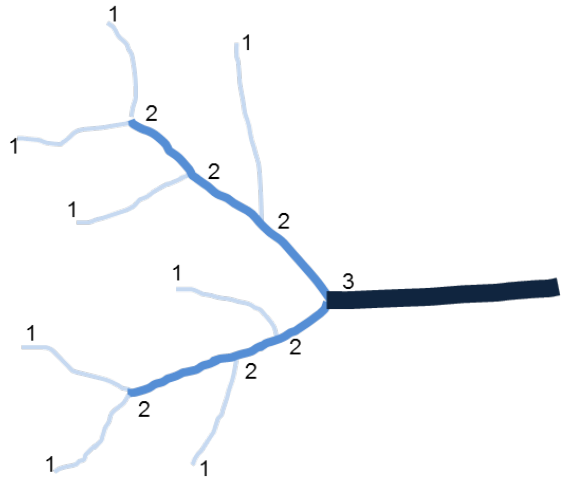


Figure 3.13.: Strahler scheme applied to a vessel tree.

This can be naturally integrated in the recursion such that the Strahler numbers are assigned in a bottom up manner. Figure 3.13 shows an example of the Strahler scheme applied to a vessel tree. It can be seen that outer branches have a lower number. This can be used to reduce a vessel tree to the inner, more important, branches on a voxel level. Because the liver vessels are usually interconnected, the Strahler scheme cannot be used during segmentation, but it could be used in other application domains, where no interconnected vessels are present (e.g. heart, lung).

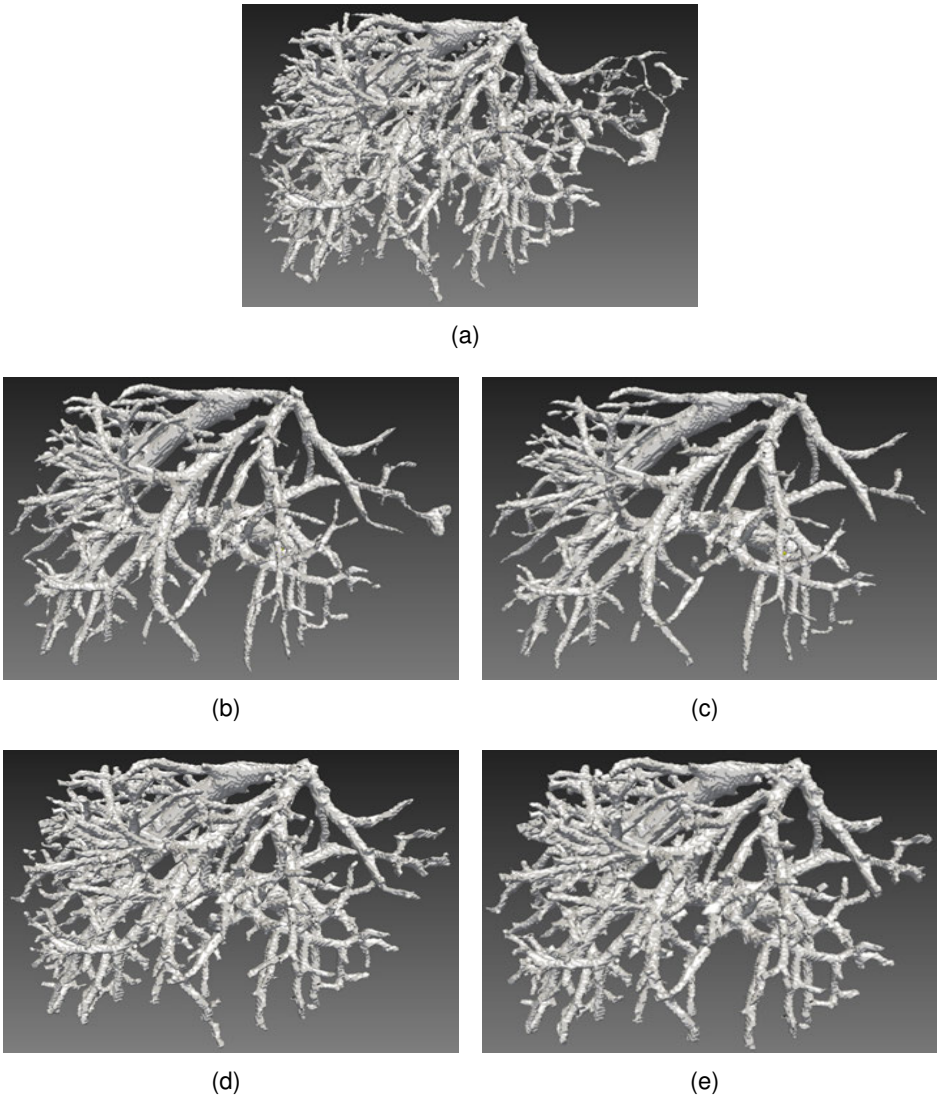
The recursion terminates when no more voxels are added to the set, denoted by $O_m = \{\emptyset\}$ for some $m \geq n+1$. The final segmentation is then given by

$$O = O_0 + O_1 + ... + O_{n+1} = \bigcup_{i=0}^{n+1} O_i. \tag{3.32}$$

Thus, a voxel is in O iff it belongs to the segmentation,

$$x \in O \Leftrightarrow S(x) = n. \tag{3.33}$$

Figure 3.14 shows a comparison between a conventional region growing and the proposed wavefront propagation technique with leak prevention. Figure 3.14(a) shows the result of a region growing algorithm. Artifacts are clearly visible on the right side. The results in Figure 3.14(b) and Figure 3.14(c) were obtained with increasing thresholds. The artifacts disappeared, but at the same time also many vessels disappeared. This is a disadvantage that is addressed with the proposed wave propagation technique. The results in Figure 3.14(d) and Figure 3.14(e) were obtained by increasing T_{leak}. It can be seen that artifacts are removed without affecting most of the vessels.



(a)

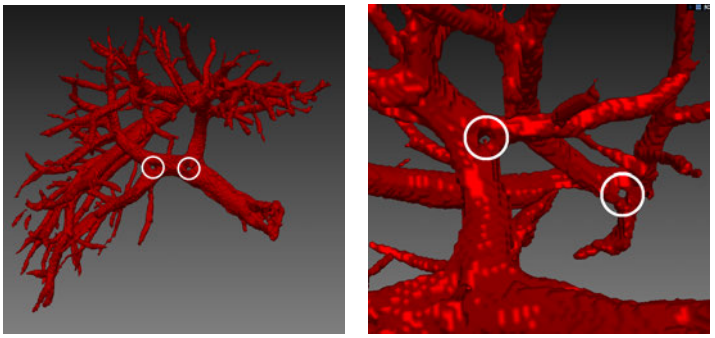
(b)

(c)

(d)

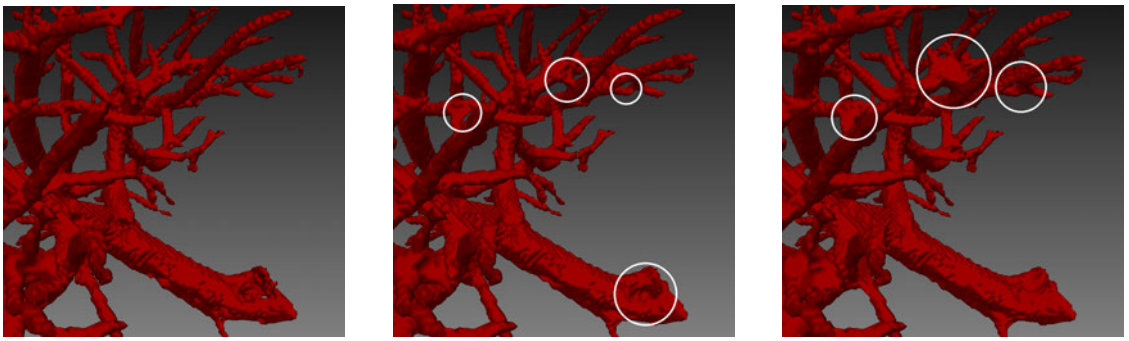
(e)

Figure 3.14.: Comparison of conventional region growing and the proposed wavefront propagation technique. (a) Result of region growing with some threshold. Artifacts are visible. (b+c) Result of region growing with increasing lower thresholds. (d+e) Result of proposed wavefront propagation with increasing T_{leak}. It can be seen that the proposed method removes the artifacts without affecting the vessels.



(a) (b)

Figure 3.15.: Varying size of holes and tunnels.



(a) (b) (c)

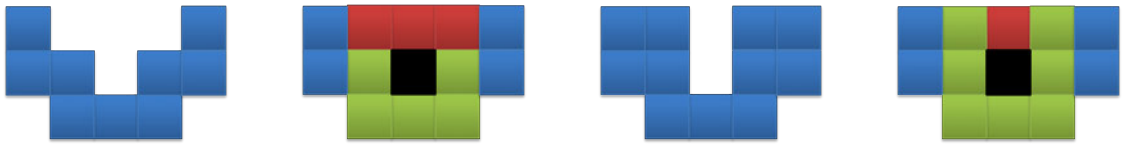
Figure 3.16.: Morphological closing to fill holes and tunnels. (a) Original image. (b) Morphological closing with ball structuring element with radius 3. (c) Morphological closing with ball structuring element with radius 5.

3.4. Volumetric hole closing

Hessian-based vessel enhancement filters are successful in identifying tubular structures, but usually have problems at junctions. Depending on the final segmentation algorithm this is visible as a hole or tunnel in the result. These defects have varying sizes, depending on the vessel diameter as can be seen in Figure 3.15.

Morphological closing is a popular method to approach this problem [CLV*10, KZZ*07]. However, morphological closing is not able to fill holes/tunnels without creating new connections or thickening the areas between branches. Figure 3.16 shows an example of morphological closing to fill holes. Figure 3.16(a) shows the original image before morphological closing was applied. Figure 3.16(b) shows the result after morphological closing with a ball structuring element with radius 3. Figure 3.16(c) shows the result after morphological closing with a ball structuring element of size 5. It can be seen in the circled areas that morphological closing creates new connections between nearby branches and thickens the areas at bifurcations. Furthermore, a large structuring element is necessary to fill all defects in the segmentation which makes the mentioned problems even worse.

The approach followed in this work is to convert background to foreground voxels based



(a) (b) (c) (d)

Figure 3.17.: Relationship between border curvature and majority threshold parameter m. (a) Relatively flat border curvature (b) The neighborhood around the current position (black) contains five foreground voxels (green) and three background voxels (red) (c) Steep border curvature. (d) The neighborhood around the current position contains seven foreground voxels and one background voxels.

on majority voting. The idea is that holes and tunnels are always embedded in the segmentation and surrounded by foreground voxels. Thus, if the neighborhood of a background voxel is enclosed by foreground voxels, it is likely that this voxel belongs to a hole and should therefore be converted to a foreground voxel. More formally this is described as follows. For a given neighborhood radius r and a majority threshold m for each background voxel it is determined if

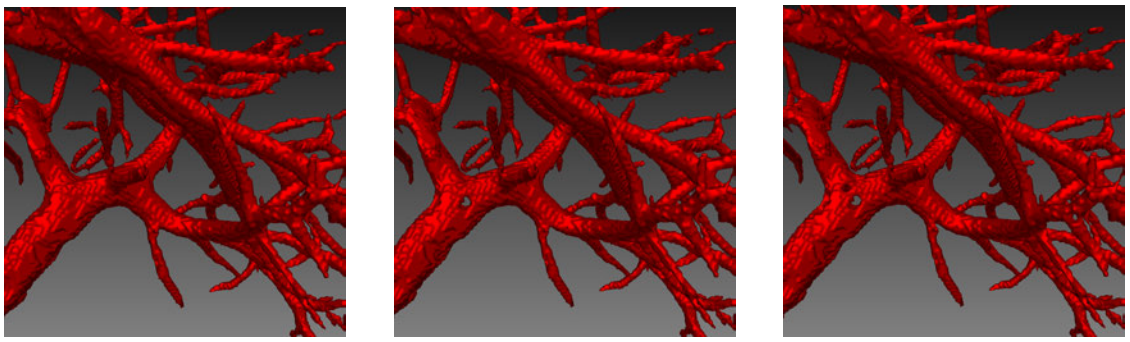
$$Vote(x, y, z, r, m, n) := \sum_{z=-r}^{r} \sum_{y=-r}^{r} \sum_{x=-r}^{r} I(x, y, z) \geq \frac{(2r+1)^3 - 1}{2} + m. \tag{3.34}$$

If this expression evaluates to true for a background voxel, then it is converted to a foreground voxel. This process is iteratively repeated n-times or until no changes occur anymore.

Two parameters control the behavior of this filter. The radius r controls the filter size and therefore, the maximum size of holes to be filled. The majority threshold m controls the influence of the curvature around the current position. Figure 3.17 illustrates this concept assuming $r = 1$. Figure 3.17(a) shows an object with a relatively flat border curvature, while Figure 3.17(c) shows an object with a steep border curvature. Foreground and background voxels are colored in green and red respectively. The current position is colored black. If $m = 1$, then in both cases, the black voxel in Figure 3.17(b) and Figure 3.17(d) would be converted to a foreground voxel. If $m = 2$, then the black voxel in Figure 3.17(b) would not meet the conversion criteria anymore, while the black voxel in Figure 3.17(d) would. The border curvature of the first object is flatter than the one of the second object. The majority threshold weights the influence of the curvature.

Figure 3.18 illustrates the influence of the majority threshold parameter in a practical example. It can be seen that the bigger the majority threshold is, the less holes are properly filled.

Another result of this method is shown in Figure 3.19. The left image shows a slice from the original image. The right image shows the same slice after applying the proposed method with $r = 1$ and $m = 1$. It can be seen that holes are properly closed without negatively affecting the shape of the vessels. As a positive side effect not only holes are eliminated, but also bays at the vessel borders. This usually results in smoother



(a) $m = 1$ (b) $m = 2$ (c) $m = 3$

Figure 3.18.: Influence of the majority threshold parameter for $r = 1$.

skeletons.

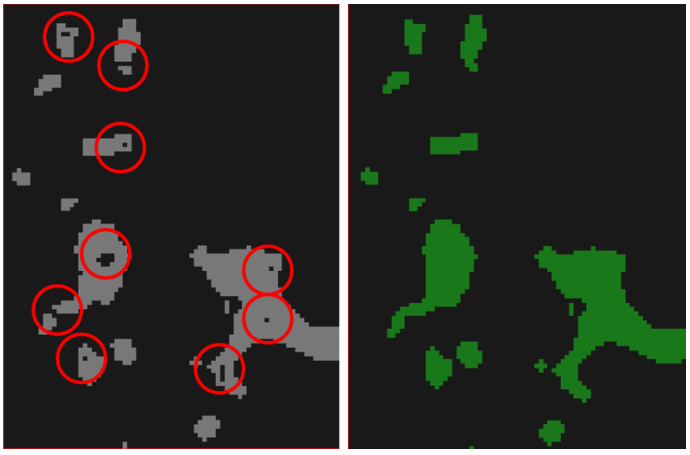
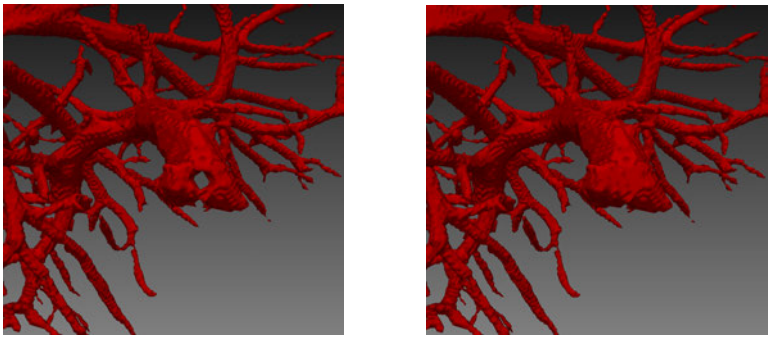


Figure 3.19.: Majority voting to fill holes and tunnels. Parameters were: $r = 1$, $m = 1$. Image taken from [DOOL11].

While this approach is able to properly fill small holes and tunnels without affecting the overall shape of the vessels, it does not properly fill big holes (see Figure 3.20(a)). On the other side, if the filter radius is just increased, then small holes can remain.

To solve for this, a two-stage cascaded hole filling as illustrated in Figure 3.21 based on majority voting is proposed. In the first stage, big holes are filled, while in the second stage remaining small holes are closed. As a matter of fact, the filter radius of the first stage has to be bigger than the one of the second stage ($r_1 > r_2$). In this work, $r_1 = 2$ and $r_2 = 1$ were used.

Figure 3.20 shows a comparison between plain majority voting and the proposed two-stage cascade. The big hole could not be filled with a single stage with parameters $r = 1$ and $m = 1$. The two-stage cascade with $r_1 = 2$ and $r_2 = 1$ was able to fill the big hole in addition to all the other smaller holes.



(a) (b)

Figure 3.20.: (a) Volumetric hole closing based on majority voting with $r = 1$, $m = 1$ and $n = 1000$ did not properly fill a big hole in this example. (b) Cascaded volumetric hole filling with $r_1 = 2$ and $m_1 = 1$ for the first stage and $r_2 = 1$ and $m_2 = 1$ for the second stage.

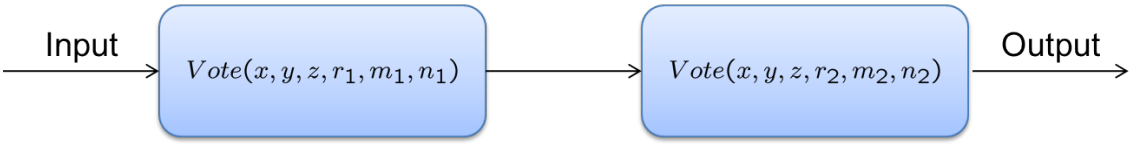


Figure 3.21.: Cascaded volumetric hole filling.

3.5. Voxel accurate graph creation

Graph representations of vessel trees have several applications in medical imaging. Surprisingly, less work about the transformation of vessel skeletons into formal graphs was published (Section 2.2.5).

In the following sections, a decomposition of vessel skeletons which simplifies the process of creating formal, attributed graphs is proposed. The basis is the well known insight that a problem is more easily solved when its complexity is spread among several abstraction layers. One key problem is the proper classification of skeleton voxels as branch voxels. Therefore, a measure to classify skeleton voxels as branch voxels is proposed in Section 3.5.1. Furthermore, the decomposition of skeletons into sub-branches as additional abstraction layer between skeleton and formal graph representation is proposed to simplify feature extraction (Section 3.5.4) and graph creation (Section 3.5.5). It will be shown in Section 3.8 that the number of branch voxels and sub-branches are not invariant under rotation and that they are normal distributed.

3.5.1. Classification of skeleton voxels

To calculate the skeleton of the segmented vessels, the thinning method by Lee et al. [LKC94] is applied (Section 2.2.5). Therefore, a freely available implementation by H. Homann was used[3]. Figure 3.22 shows the segmented portal vein (transparent red) and the corresponding skeleton in black, both visualized as isosurfaces. Please note the bad

[3] http://hdl.handle.net/1926/1292

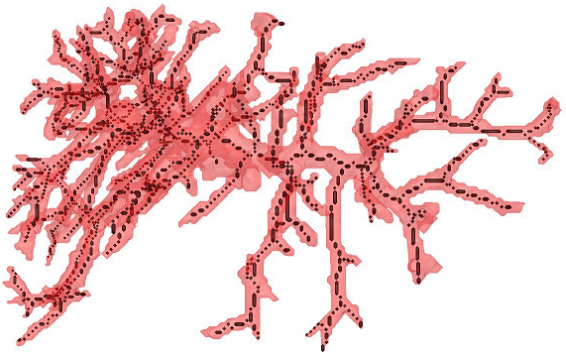
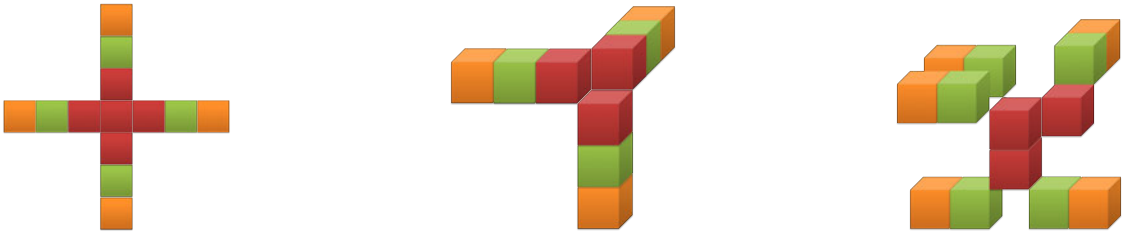


Figure 3.22.: Segmented portal vein (red) and its skeleton (black).



(a) (b) (c)

Figure 3.23.: Examples of ambiguous branch voxel classifications. Orange: End voxels. Green: Regular voxels. Red: Branch voxels. Image taken from [DOL10c].

quality of the skeleton visualization and compare it with the results in Section 3.8.

To prepare the skeleton for its transformation into a graph, its voxels must be classified into regular voxels, end voxels and branch voxels. Branch and end voxels will be represented by nodes in the graph, while regular voxels will be grouped and represented by edges. In the following, $\vec{v}$ denotes the position of a voxel in index coordinates. The position of a voxel in physical space is denoted by $Physical(\vec{v})$. The intensity value of a voxel in the original image at position $\vec{v}$ is denoted by $Intensity(\vec{v})$ or just v.

The 26-neighborhood of a voxel is defined as all its surrounding visible voxels, denoted by

$$N_{26}(\vec{v}) = \{\vec{w} | \max_{i \in \{0,1,2\}} (|\vec{v}_i - \vec{w}_i|) = 1 \wedge w = 1\} \tag{3.35}$$

where $\vec{v}_i$ is the i-th component (x,y,z) of the vector. Typically, a voxel is classified as end voxel if $|N_{26}(\vec{v})| = 1$, as regular voxel if $|N_{26}(\vec{v})| = 2$ and as branch voxel if $|N_{26}(\vec{v})| \geq 3$. However, the classification into branch voxels is ambiguous for several cases of which some are shown in Figure 3.23. The red voxels are possible branch voxels.

The question is now, how to pick the good one? Several solutions to this problem were published, but they fail for one case or the other (see section 2.2.5). This problem is solved in several steps. Therefore, branch candidate voxels are introduced. That is, voxels for which $|N_{26}(\vec{v})| \geq 3$, but are not classified as branch voxels yet. Next, the branch candidate neighbors of a voxel are defined as

$$C(\vec{v}) = \{\vec{w} | \vec{w} \in N_{26}(\vec{v}) \wedge |N_{26}(\vec{w})| \geq 3\}. \tag{3.36}$$

Then, for every branch candidate voxel $\vec{v}$ a measure is calculated that is higher the more branch candidate neighbors it has and the closer they are. This is expressed as

$$f(\vec{v}) = |C(\vec{v})| \sum_{\vec{w} \in C(\vec{v})} (3 - \|\vec{v} - \vec{w}\|^2 + 1). \tag{3.37}$$

where $\|\cdot\|$ is the euclidean norm and $|N_{26}(\vec{v})| \geq 3$. Among all evaluated branch candidate voxels within a 26-neighborhood, local maxima are searched and converted to branch voxels. This is denoted by

$$B = \underset{\vec{x} \in \{\vec{v}\} \cup C(\vec{v}), \left|N_{26}(\vec{v})\right| \geq 3}{\operatorname{argmax}} f(\vec{x}) \tag{3.38}$$

Several datasets were analyzed and it was found that patterns with two and three branch voxels due to symmetrically arranged voxels can occur as shown in Figure 3.24 (a) and (b). From a graph-creation point of view two neighboring branch voxels are easily handled by connecting them through an edge. However, three neighboring branch voxels cause an ambiguous situation and must be resolved to avoid circles in the graph which are topologically equivalent to a hole. A possible way to handle this situation is to favor branch candidate voxels that are closer to the root of the tree as Palágyi et al. did [PTHS06]. However, this implies that it is known where the root is, which is not the case at this step without user interaction and the result may not be worth the additional effort. This case is approached by traversing the whole image. For each background voxel, it is checked if there are three branch voxels in the 26-neighborhood that form a pattern like the one in Figure 3.24 (b). Due to symmetry reasons, there are 24 equivalent patterns in 3D that need to be checked. To find these patterns efficiently, each 26-neighborhood is encoded and interpreted as a bit string and natural number respectively. Using a sequential numbering scheme, it follows that

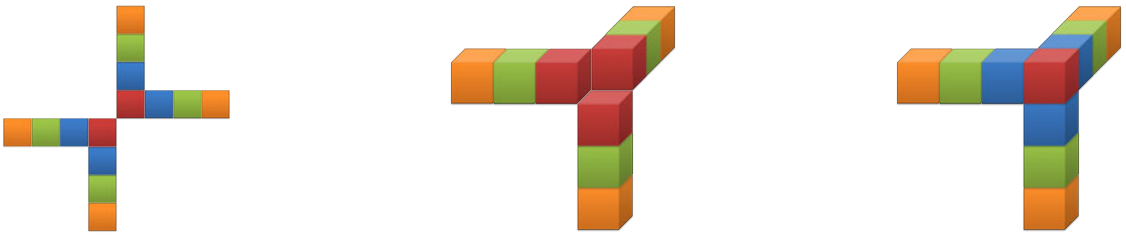
$$V_{26}(\vec{v}) = v_{26} v_{25} \ldots v_1 v_0 = \sum_{i=0}^{26} v_i \cdot 2^i \tag{3.39}$$

where $\vec{v}$ is the voxel in the center, $v_i \in [0, 1]$ and $V_{26} \in \mathbb{N}$. The set of patterns $P = \{P_0, \ldots, P_{23}\}$ to be recognized are encoded using the same method. For example, the three red branch voxels in Figure 3.24 (b) are encoded as

$$\begin{aligned} P_0 &= 000000010 \quad 000010001 \quad 000000000 \\ &= 2^{19} + 2^{13} + 2^9 = 532992. \end{aligned}$$

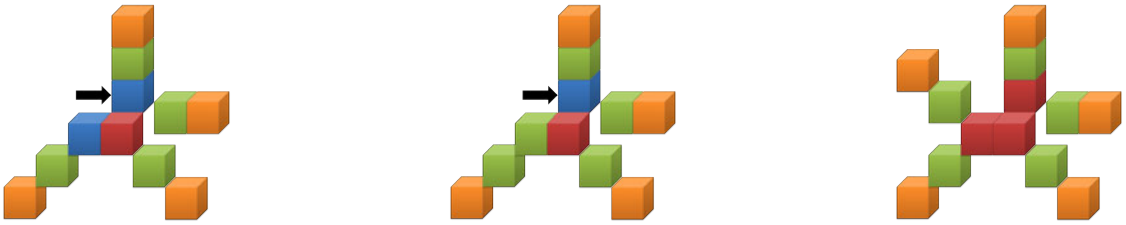
If a background voxel $\vec{v}$ is found for which $|N_{26}(\vec{v})| = 3$ and $V_{26}(\vec{v}) \in P$, a branch voxel is created at its position and the other branch voxels are degraded to branch candidate voxels (Figure 3.24 (b) and (c)).

In the next step, the remaining branch candidate voxels are handled. As can be seen in Figure 3.25 (a), among the remaining branch candidate voxels are candidates that have two neighboring regular voxels (black arrow). They act as branch voxels, but are not marked as branch voxel to avoid ambiguous situations like the one shown in Figure 3.25 (c). Thus, branch candidate voxels with one regular voxel in their 26-neighborhood are



(a) (b) (c)

Figure 3.24.: (a) Voxel classification with two neighboring branch voxels (red) due to symmetric voxel arrangements. (b) Voxel classification with three neighboring branch voxels (red) due to symmetric voxel arrangements. (c) A branch voxel (red) was added and previous branch voxels were degraded to branch candidate voxels (blue). Image taken from [DOL10c].



(a) (b) (c)

Figure 3.25.: (a) After classification of branch voxels (red) there can be branch candidate voxels (blue) that are root voxels of a subtree with at least two branches (black arrow). (b) Branch candidate voxels with only one regular voxel in its neighborhood are classified as regular voxel. (c) Ambiguous situation if branch candidate voxels with two neighboring regular voxels are converted to branch voxels. Image taken from [DOL10c].

converted to regular voxels (Figure 3.25 (b)) or to an end voxel, depending on how many non branch candidate neighbors they have.

3.5.2. Skeleton decomposition into sub-branches

After each skeleton voxel has been properly classified, the skeleton is decomposed into sub-branches which are separately stored. This step abstracts from the image data structure and stores each voxel together with its physical and index coordinates in an easy to access data structure like e.g. a vector. It is an intermediate representation between an image and a formal graph. This greatly simplifies further processing of the skeleton in the next steps.

Let $W = (\vec{v}_0, \vec{v}_1, \ldots, \vec{v}_{n-1})$ be a sequence of skeleton voxels, $n \in \mathbb{N}$. W is called a sub-branch of length n if

- $\vec{v}_0$ is a branch voxel,
- $\vec{v}_{n-1}$ is a branch or end voxel,

Figure 3.26.: Unpleasant visualization of a skeleton using marching cubes.

- $\vec{v}_i \in N_{26}(\vec{v}_{i-1})$, $0 < i \leq n-1$
- $\vec{v}_k$ is a branch voxel $\Rightarrow k = n-1, k \geq 1$

Figure 3.29 (a) is an example of a skeleton that can be decomposed into four sub-branches (Figure 3.29 (b)).

To decompose the skeleton, all branch voxels are processed and voxels connected to it followed according to the statemachine shown in Figure 3.30. If the current position is a branch candidate voxel, a splitting takes place. The number of sub-branches created during this splitting step depends on the actual situation in the neighborhood. The possible choices and transitions are drawn with a dashed line in Figure 3.30.

3.5.3. Skeleton visualization

Visualizing a skeleton in 3D is not trivial. The marching cubes method [LC87] is widely used to create a polygonal mesh of an isosurface. In case of a skeleton, it produces very unpleasant results as shown in Figure 3.26.

The decomposed skeleton can also be used to create appealing visualizations of the skeleton. Therefore, all elements in a sub-branch are connected as piecewise polylines. A result is shown in Figure 3.27(a).

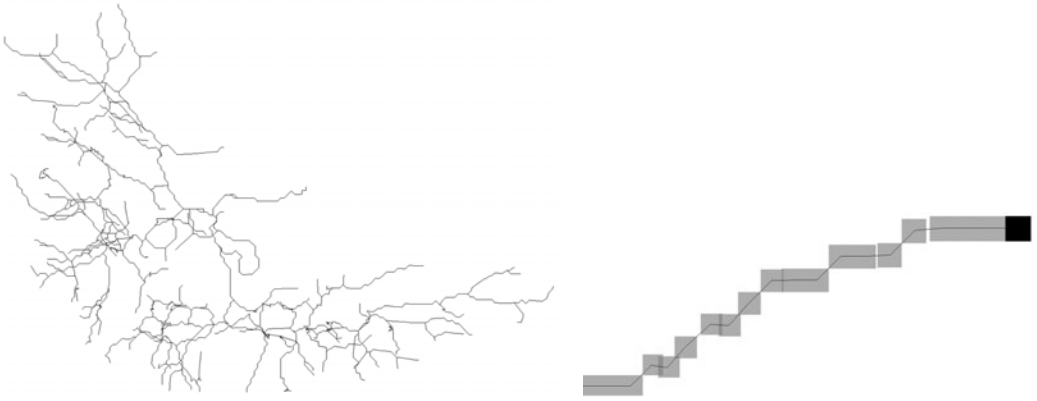
It can be seen that the visualization of the skeleton has improved. However, the skeleton looks bumpy. Figure 3.27(b) shows a close-up of the skeleton, which shows that the discrete nature of the skeleton is the cause.

To smooth the skeleton, a central moving average filter of rank 3 is applied to each element of a sub-branch in physical space. Formally,

$$Physical(\vec{v}_i) = \frac{1}{3}(Physical(\vec{v}_{i-1}) + Physical(\vec{v}_i) + Physical(\vec{v}_{i+1})) \tag{3.40}$$

with $i = 1, 2, \ldots, n-1$.

Figure 3.28(a) shows the skeleton after applying the smoothing filter and Figure 3.28(b) shows a close-up of a branch. It can be seen that the visualization of the skeleton is much smoother now.



(a) Skeleton polyline (b) Close-up

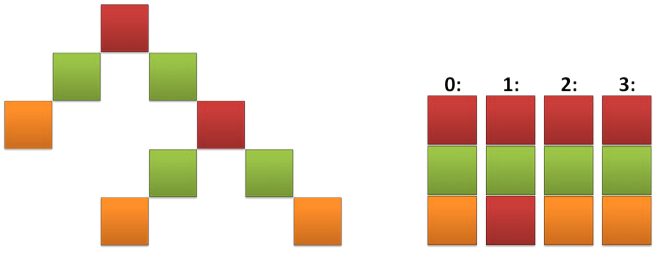
Figure 3.27.: Skeleton visualization using polylines. (a) Result after connecting all elements in a sub-branch with piecewise polylines. (b) Close-up of a branch.

(a) Smoothed polyline (b) Close-up

Figure 3.28.: (a) Result after smoothing each sub-branch with a central moving average filter of rank 3. (b) Close-up of a branch.

3.5.4. Feature extraction

In the following, some features are described that can be very easily calculated using the proposed skeleton decomposition scheme. They are stored in a higher-order data



(a) (b)

Figure 3.29.: (a) A skeleton after voxel classification. (b) The same skeleton decomposed into sub-branches. Image taken from [DOL10c].

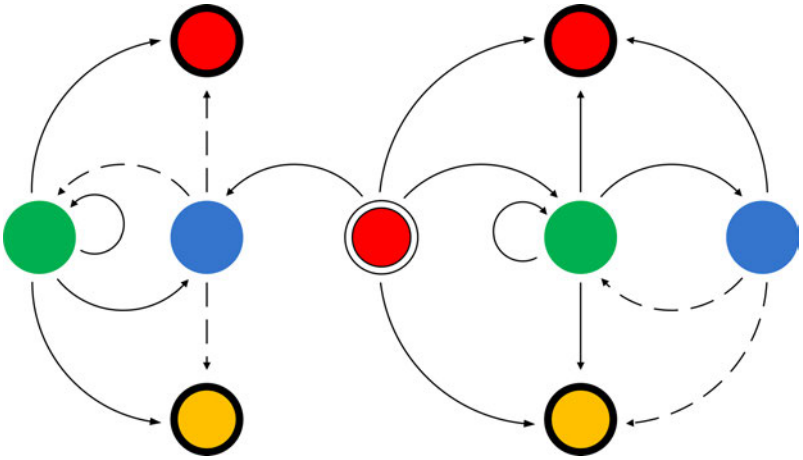


Figure 3.30.: A state machine used to traverse and decompose a skeleton with classified voxels. Red: Branch voxel, Blue: Branch candidate voxel, Green: Regular voxel, Orange: End voxel. Circles with thick black borders are stop states. Circles with thick white border are start states. Image taken from [DOL10c].

structure and later added as attributes to the final graph.

3.5.4.1. Statistical measures

The mean and standard deviation of the intensity values of a sub-branch W is calculated as

$$\overline{Mean} = \frac{1}{n}\sum_{k=0}^{n-1} Intensity(\vec{v}_k) \tag{3.41}$$

and

$$SD = \sqrt{\frac{1}{n-1}\sum_{k=0}^{n-1}(Intensity(\vec{v}_k) - \overline{Mean})^2} \tag{3.42}$$

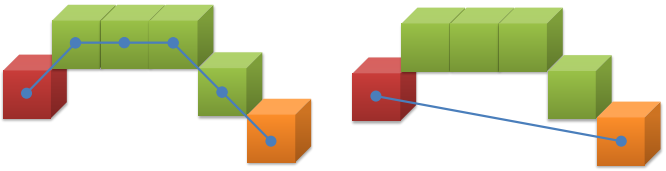
respectively. This measure was successfully used in one contrast enhanced CT dataset of the liver to detect and remove the liver artery, which was brighter than the portal vein and responsible for a loop in the graph.

3.5.4.2. Length

The length of a sub-branch is the summation of the physical distances between two voxels, denoted by

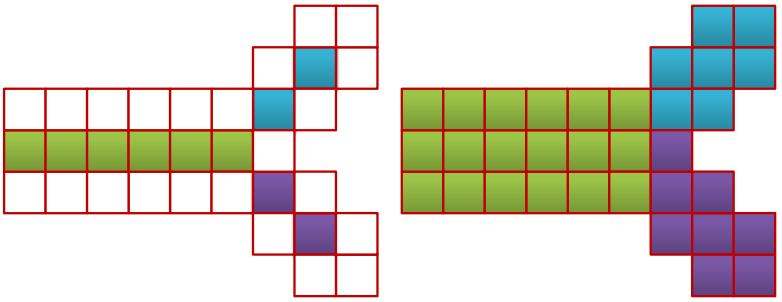
$$Length = \sum_{i=0}^{n-2} \|Physical(\vec{v}_i) - Physical(\vec{v}_{i+1})\|. \tag{3.43}$$

This is illustrated in Figure 3.31(a).



(a) (b)

Figure 3.31.: (a) Calculation of the length as distance between neighboring voxels. (b) Calculation of the distance as the Euclidean distance between start and end voxel.



(a) (b)

Figure 3.32.: (a) Segmented vessels (outlined) and corresponding sub-branches (solid). (b) After a nearest neighbor search the vessel segmentation is filled with the same value as the closest sub-branch.

3.5.4.3. Distance

The distance of a sub-branch is defined as the physical distance between start and end voxels.

$$Distance = \|Physical(\vec{v}_{n-1}) - Physical(\vec{v}_0)\| \tag{3.44}$$

This is illustrated in Figure 3.31(b).

3.5.4.4. Volume

To calculate the volume of a sub-branch, the nearest neighbor approach is used for which the vessel segmentation is needed. For every voxel $\vec{v}$ of the segmented vessels, the nearest (with respect to the Euclidean norm) sub-branch is searched and $\vec{v}$ is labeled with a value that is associated with that sub-branch. Figure 3.32 shows an example of this process. Now, let N be the number of voxels belonging to a sub-branch, then the volume is calculated as

$$Volume = N \cdot vol_v \tag{3.45}$$

where vol_v is a constant specifying the volume of a voxel.

The usage of the decomposed skeleton has two advantages. First, the fact can be used that each sub-branch of the decomposed skeleton has implicitly a unique ID. This is because each sub-branch is stored in a vector, which is sequentially numbered. Second, to find the nearest sub-branch, the whole image does not have to be searched, but only

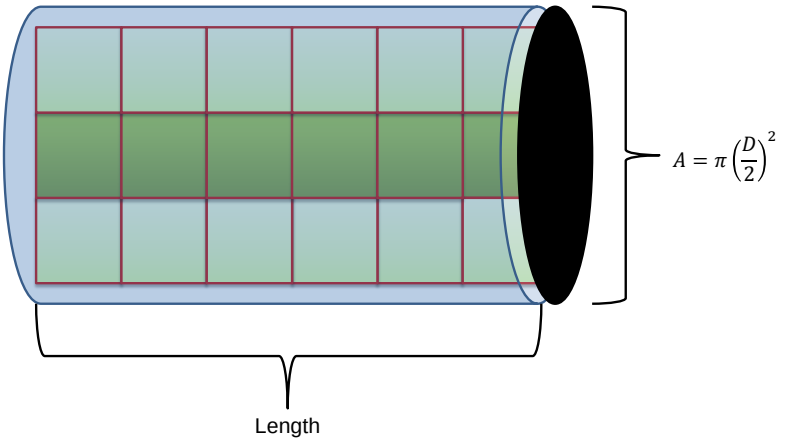


Figure 3.33.: Calculation of the average diameter.

the actual skeleton voxels. Thus, it is very efficient.

3.5.4.5. Average diameter

As shown in Figure 3.33, the average diameter of a sub-branch can be calculated as

$$Diameter_{avg} = 2 \cdot \sqrt{\frac{Volume}{\pi \cdot Length}}. \tag{3.46}$$

This is only an approximation, because it assumes that a sub-branch has a cylindrical shape, which is not always true.

3.5.4.6. Curveness

The curveness of a sub-branch is defined as the deviation from a straight line. It can be calculated as

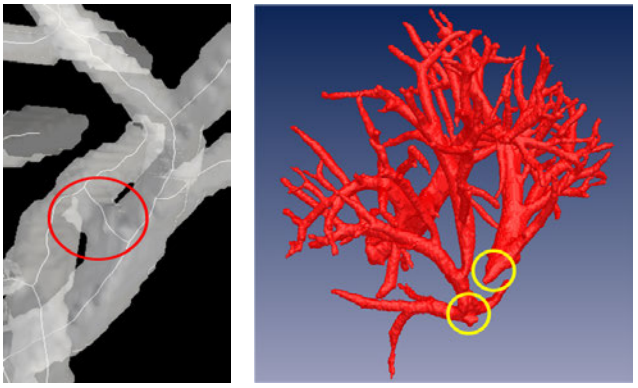
$$Curveness = \frac{Length}{Distance}. \tag{3.47}$$

3.5.5. Graph creation

An undirected graph is an ordered pair $G = (V, E)$ comprising a set of vertices $v \in V$ and edges $\{u, v\} \in E$ where $u, v \in V$. The goal is to find a mapping of sub-branches $W = (\vec{v}_0, \vec{v}_1, \ldots, \vec{v}_{n-1})$ to the graph G. The idea is that each sub-branch is an edge in the formal graph. Start and end voxels of a sub-branch are vertices in the graph.

Let S be the set of all decomposed sub-branches. First, a set is built which contains a node for every starting voxel of all sub-branches as

$$V_1 = \{v | W \in S.\vec{v} \in W.\forall \vec{w} \in W.(\vec{w}, \vec{v}) \notin W\}. \tag{3.48}$$



(a) (b)

Figure 3.34.: Segmentation imperfections. (a) Interconnected branches of different vessel systems. (b) Splitted Vessel tree. Image taken from [DOOL11].

In the same way, for all end voxels of all sub-branches a set is built as

$$V_2 = \{v | W \in S.\vec{v} \in W.\forall \vec{w} \in W.(\vec{v}, \vec{w}) \notin W\}. \tag{3.49}$$

Edges belong to the graph, if their start and end nodes correspond to start and end voxels of the same sub-branch. This is expressed as

$$E = \{\{u, v\} | W \in S.u \in V_1 \wedge v \in V_2 \wedge \vec{u}, \vec{v} \in W\}. \tag{3.50}$$

The graph is then defined as $G = (V_1 \cup V_2, E)$.

3.6. Graph-based separation of hepatic veins

The proper separation of hepatic veins is crucial for discriminative visualization during surgical planning and intraoperative navigation, automated analysis and liver segment approximation. However, due to low resolutions, partial volume effects, unequal distributed contrast agent, motion artifacts and imperfect segmentation algorithms it happens that portal vein and liver vein appear to be connected at some points (e.g. Figure 3.34(a)). Furthermore, a vessel system can be split into multiple parts (Figure 3.34(b)).

A weakness of current state of the art methods is that they do not take into account the differences between the vessel structures in a real physical liver and the mapping of these structures using imaging algorithms into a discrete volume. For example, in experiments the diameter of a branch was not always decreasing. Figure 3.35 shows an example taken from a real dataset. It can be seen that the diameter of a following branch is not constantly decreasing, but rather unsteady. This is usually due to noise and imperfect segmentation algorithms. Furthermore, it happened that obtuse angles at branches were present without being erroneous. In general, it was found that the violation of one constraint is not always enough to decide if a branch is erroneous.

In the following sections, a novel method to separate interconnected hepatic vessel systems is proposed. It handles several imperfections by taking the number of constraint

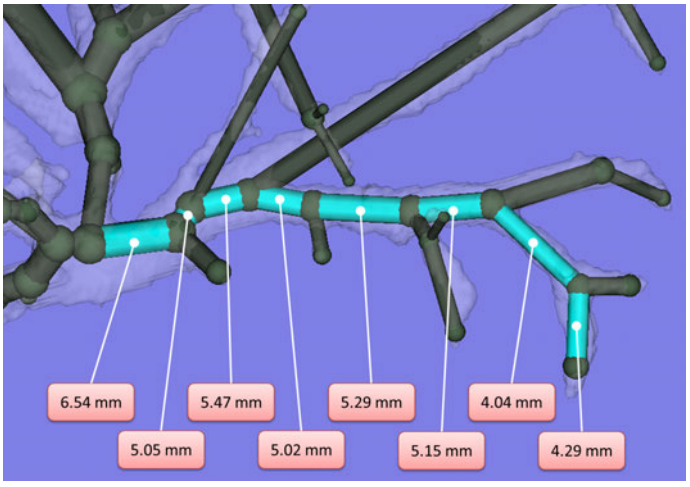


Figure 3.35.: Diameters of branches along a path are unsteady, but in general diameters tend to decrease.

Edge	a	b	c	d	e	f	g
Diameter	10.1	8.3	6.5	4.7	6.9	8.1	10.2

Table 3.3.: Diameters of the edges on the path between the two root nodes in Figure 3.36.

violations per branch into account. Furthermore, similar to the algorithm proposed by Göpfert et al. [GGD*98], it assumes that wrong connections can only occur on paths between root nodes of different vessel systems. It is automatic in the sense that a separation suggestion is generated fully automatically. It is semi-automatic in the sense that the user has to provide the locations of root nodes and the user can manually separate both trees efficiently and ask for computer assistance at any time or modify automatically generated suggestions. Furthermore, it is shown how to transfer the graph-based separation results back to the imaging data.

3.6.1. Main algorithm

The general idea of the proposed algorithm is described as follows. An error counter is assigned to every edge of the graph, which will be incremented or decremented depending on the location, orientation and shape of the underlying vessel branch. Only edges are taken into account that are on paths between root nodes of different vessel systems. Then, to separate the vessels, each path is inspected separately and the edge with the highest error counter is removed. This procedure is repeated until no connecting paths between both trees exist. Formally this is described as follows.

Let $G = (V, E, R \subset V)$ be an undirected graph with edges $e = \{u, v\} \in E$, vertices $v \in V$ and root nodes R. Furthermore, $d(e)$ returns the diameter, $l(e)$ the length and $adj(e)$ adjacent edges of e. E_{path} is the set of edges that are on paths between root nodes. C is the set of edges on cycles. $(e_0, ..., e_n) \in A_t$ are edge vectors of already traversed (sub-)paths at iteration step t. J is a set of edges where two paths merge during traversion. E_{rest} is the set of edges to be processed and E_{vis} is the set of edges that are already processed.

At the beginning ($t = 1$), E_{rest}, E_{vis} and A_1 are initialized as follows.

$$E_{rest} = E_{path} \tag{3.51}$$
$$E_{vis} = E \setminus E_{path} \tag{3.52}$$
$$A_1 = \{\{v_0, v_1\} \mid \{v_0, v_1\} \in E_{rest} \wedge (v0 \in R \vee v1 \in R)\}. \tag{3.53}$$

Next, the path

$$a_{max} = (e_0, ..., e_n) = \underset{(e_0,...,e_n)\in A_t}{\operatorname{argmax}} d(e_n) \tag{3.54}$$

is determined, where the last edge has the largest diameter among all paths in A_t. This edge is removed from E_{rest} and added to E_{vis}:

$$E_{vis} = E_{vis} \cup \{e_n\} \tag{3.55}$$
$$E_{rest} = E_{rest} \setminus \{e_n\} \tag{3.56}$$

Vessels of different vessel systems are usually not interconnected near root nodes. The probability for an interconnection increases the farther away an edge is. Thus, the error counter for an edge on path a_{max} is updated if

$$\sum_{i=0}^{n} l(e_i) \leq S_{pl}, \tag{3.57}$$

where S_{pl} is an experimentally determined threshold.

Next, the model assumptions are verified for the path a_{max}. The conditions when the error counter is in/descreased are explained in Section 3.6.2.

If two paths merge at edges e_m and f_m, they are stored in set J. Formally, J is extended as follows.

$$J = J \cup \{\{e_m, f_m\} \mid (f_0, ..., f_m) \in A_t \wedge e_m \in a_{max} \wedge e_m \in adj(f_m)\}. \tag{3.58}$$

For the next iteration a_{max} is removed from A_{t-1} and new paths are added as follows:

$$A_t = A_{t-1} \cup \{(e_0, ...e_n) \mid (e_0, ..., e_{n-1}) = a_{max} \wedge e_n \in adj(a_{n-1}) \wedge e_{n-1} \notin adj(e_{n-2}) \wedge e_n \notin E_{vis}\}. \tag{3.59}$$

These steps are repeated until A_t is empty. Figure 3.36 shows an example of the described traversal procedure. The assumed diameters for the edges $a - g$ are listed in Table 3.3.

Afterwards the edges to be removed can be determined. Therefore, the shortest path between two root nodes is calculated and the edge with the highest error counter value is removed. If more than one exists, then the one with the smallest diameter is deleted. In the case that all error counters on a path are unchanged, Selle's approach is used and the edge with the lowest diameter that is also stored in J is deleted. This has to be repeated until no paths between two root nodes exist anymore. Formally this is described as follows.

$spath(u, v)$ returns the shortest path between root nodes u and v if it exists. $error(e)$

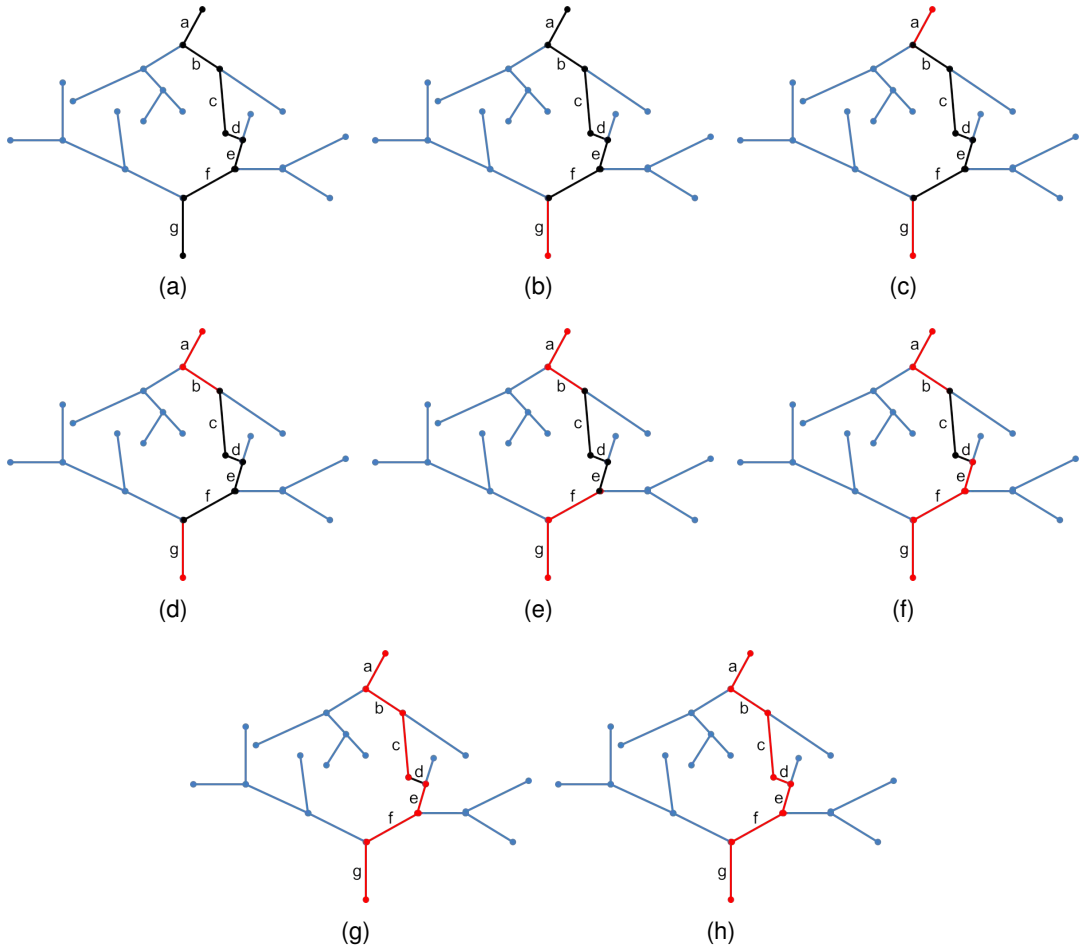


Figure 3.36.: Traversal of two interconnected trees. Only edges between root nodes are taken into account and traversal order depends on the diameter of the last edge.

returns the error counter value of edge e. Let

$$a_{con} = (e_0, ..., e_n) = spath(r_i, r_j) \tag{3.60}$$

be the shortest path between r_i and r_j, $i \neq j$ and $r_i, r_j \in R$. The set of edges that can be potentially removed is determined as

$$E_{rem} = \{e \in a_{con} \mid \forall f \in a_{con}.error(e) \geq error(f) > 0\}. \tag{3.61}$$

If $\|E_{rem}\| = 0$, then J is used to determine potentially removable edges as

$$E_{rem} = \{e, f \in a_{con} \mid \{e, f\} \in J \vee \{f, e\} \in J\}. \tag{3.62}$$

Afterwards, the edge with the smallest diameter is determined as

$$e_{rem} = \underset{e \in E_{rem}}{\operatorname{argmin}}\, d(e) \tag{3.63}$$

and removed from E.

3.6.2. Detecting violations of model assumptions

A deviation from the following assumptions will increment or decrement the error counter of an edge.

The vessel thickness decreases in flow direction. Figure 3.35 showed that the thickness of branches is unsteady along a path. However, the tendency is that it is decreasing. To account for that, edges e_{suc} and e_{pred} are determined with the largest diameters that are before and after the last edge in a_{max} within maximal S_{suc} and S_{pred} edges distance. The error counter of the edge with the smallest diameter is then incremented.

Edges after the last edge of a_{max} are determined as follows:

$$E_{suc} = \{e \mid e_{n+1} \in E_{path} \wedge e_{n+1} \in adj(e_{n+i-1}) \wedge e_{n+i} \neq adj(e_{n+i-2})\}. \tag{3.64}$$

The edge with the largest diameter is then found as

$$e_{suc} = \underset{e \in E_{suc}}{\operatorname{argmax}}\, d(e). \tag{3.65}$$

The edges before the last edge of a_{max} are determined as

$$E_{pred} = \{e \mid e_i \in a_{max} = (e_1, .., e_n) \wedge n - S_{pred} \geq i \leq n\} \tag{3.66}$$

and the edge with the largest diameter as

$$e_{pred} = \underset{e \in E_{pred}}{\operatorname{argmax}}\, d(e). \tag{3.67}$$

Let

$$a_{con} = (e_{pred}, ..., e_n, ...e_{suc}) \tag{3.68}$$

be the path between e_{pred} and e_{suc}. Now, if $d(e_{suc}) \geq d(e_{pred}) + S_{diam}$ then the error counter of

$$e_{weak} = \underset{e \in a_{con}}{\operatorname{argmin}}\, d(e) \tag{3.69}$$

is incremented. To ensure that an edge is only punished once for a delict, e_{weak} is stored in another set E_{diam}.

It exists no obtuse angle between two successive branches. Figure 3.37 shows a part of a vessel system extracted from a clinical dataset. It can be seen that bifurcation (α and β) angles are acute in the direction from root to leaf. To account for that, edges E_{angle} are determined that are adjacent to the last edge e_n in a_{max}, but not to e_{n-1}. Each edge is then converted to a directed edge such that e_n points towards the edges in E_{angle} and edges in E_{angle} point away from e_n. Let $e = (\vec{u_1}, \vec{u_2})$ and $f = (\vec{v_1}, \vec{v_2})$ be two directed edges, then the angle between them is calculated as

$$angle(e, f) = \frac{(\vec{u_2} - \vec{u_1}) \cdot (\vec{v_2} - \vec{v_1})}{\|\vec{u_2} - \vec{u_1}\| \cdot \|\vec{v_2} - \vec{v_1}\|}. \tag{3.70}$$

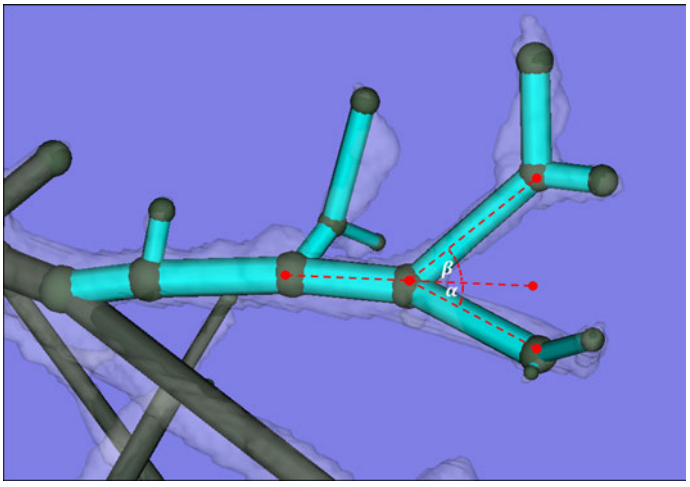


Figure 3.37.: Bifurcation angles are usually not obtuse.

If

$$\underset{e \in E_{angle}}{\operatorname{argmax}}\, angle(e_n, e) \geq S_{Angle} \tag{3.71}$$

then the error counter of e_n is incremented. The error counter for e is also incremented, if $e \in E_{path}$.

Vessels are tree-shaped and no loops exist. To account for that, the error counter for $e_n \in a_{max}$ is incremented, if $e_n \in C$.

The curvature does not deviate from an average value. This is related to the angle between two edges. However, cases exist where the tips of two vessel branches seem to be connected and are represented by only one edge in the graph. To account for this,

$$dist(e = (u, v)) = v - u \tag{3.72}$$

is defined to be the distance from start to end vertices of an edge in the graph and $len(e)$ the length of the voxel skeleton. The deviation from a straight branch is calculated as

$$curv(e) = \frac{len(e)}{dist(e)}. \tag{3.73}$$

If

$$curv(e_n) \geq S_{curv}, \tag{3.74}$$

$e_n \in a_{max}$, then the error counter for e_n is incremented.

Main branches form a straight line or are just slightly curved. Branches of the same tree usually run approximately in the same direction or are just slightly curved. At interconnection points this seems not to be true. To account for this, a set of directed edges is determined and created as

$$E_{from} = \{(u_1, u_2) = e \mid u_2 = v_1 \wedge \{u_1, u_2\} \in adj(e_n) \wedge angle(e, e_n) < S_{path}\} \tag{3.75}$$

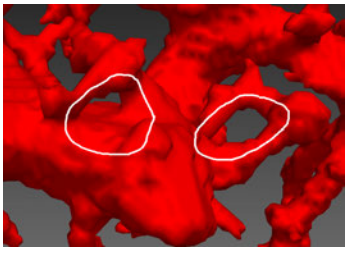


Figure 3.38.: Loops due to liver arteries. Image taken from [DOOL11].

which are adjacent to $e_n = (v_1, v_2)$ and point away from it and a set of directed edges

$$E_{to} = \{(u_1, u_2) = e \mid u_1 = v_2 \wedge \{u_1, u_2\} \in adj(e_n) \wedge angle(e, e_n) < S_{path}\} \tag{3.76}$$

which are adjacent towards e_n and point to it. If $\|E_{from}\| \neq 0$ and $\|E_{to}\| \neq 0$ then decrement the error counter for all edges in E_{from} and E_{to} and for $e_n \in a_{max}$. To decrement their error counters only once during this step, these edges are stored in a set $K_{straight}$.

3.6.3. Handling cycles and loops

Two possible causes for cycles within a vessel system were identified. The first cause is due to segmentation algorithms that use e.g. vesselness filters as the basis. These filters are successful in identifying tubular structures, but usually have problems at junctions. Depending on the final segmentation algorithm this is visible as a hole or tunnel in the final result. A hole/tunnel is mapped to a cycle in the formal graph. Volumetric hole closing was already discussed in Section 3.4. A second cause of loops in a graph are the liver arteries. They have small diameter and usually follow the portal vein very closely. However, sometimes the arteries form a loop as shown in Figure 3.38. To open these loops a simple heuristic is applied which is described in the next paragraph.

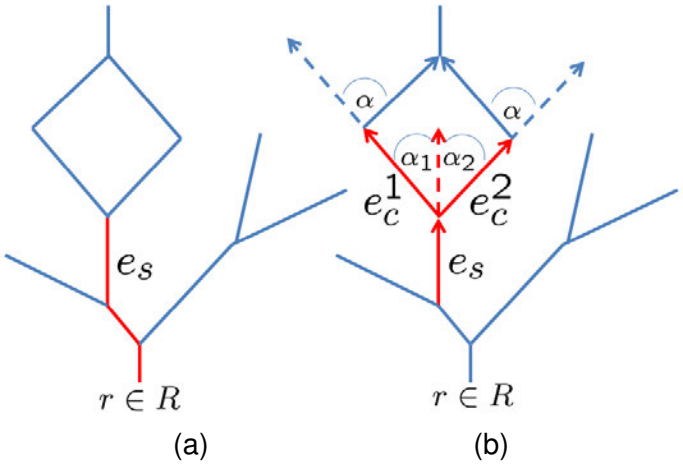
Removal of Liver Arteries. Let $G_n = (V_n, E_n, R_n \subset V_n)$ be the n separated graphs of portal and liver vein with edges $e = (u, v) \in E_n$, vertices $v \in V_n$ and root nodes $r \in R_n$. $e_c \in C_n$ are edges on cycles. Traversal of the graphs starts at each root r until an entrance edge $e_s = (u, v) \in E_n, v \in e_c \in C_n$ into a loop is found (Figure 3.39(a)). Next, the angles α_i between e_s and following edges e_c^i (Figure 3.39(b)) are measured. The maximum angle and the corresponding edge are stored in e_{max} and α_{max}, respectively:

$$e_{max} = \underset{e_c^i}{\operatorname{argmax}}\, angle(e_s, e_c^i) \tag{3.77}$$

and

$$\alpha_{max} = \max_i angle(e_s, e_c^i). \tag{3.78}$$

Traversal of the paths is continued in parallel and α_{max} and e_{max} are updated accordingly. Within the cycle e_{max} is set to the edge with the smaller diameter. If the cycle is traversed e_{max} is removed.



(a) (b)

Figure 3.39.: (a) Entrance edge into a cycle. (b) Angle measurements. Image taken from [DOOL11].

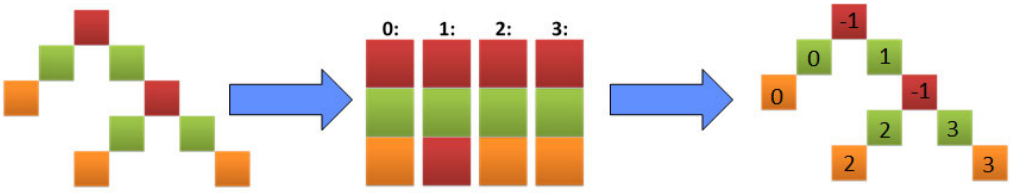


Figure 3.40.: Decomposition of vessel skeletons into sub-branches to assign unique labels to each sub-branch. Image taken from [DOL10a].

3.6.4. Transfer of separation results to image data

After separation of the mathematical graph into portal and hepatic veins, segmentations and skeletons are still not separated. Separated vessel segmentations are used e.g. for visualizing both vessel systems overlapped in 2D slices in different colors and for an accurate 3D visualization. The skeleton of the portal vein and its mathematical graph representation is needed to e.g. calculate patient specific functional segments following the Couinaud scheme. This section describes a method to transfer the graph-based separation results to the image data.

3.6.4.1. Separating skeletons

Theoretically, this step could be skipped and only the segmentations could be separated (Section 3.6.4.2) and skeletons and formal graphs could be calculated from these segmentations again. However, it is more efficient (read: faster) to use the available skeletons and separate them. Therefore, during pre-processing as explained in Section 3.5.2, skeletons are decomposed into sub-branches. Each sub-branch gets a unique ID. This ID is used to label the skeleton voxels in the image data accordingly. Branch voxels represent a special situation and get a special label (e.g. a constant tag like '-1'). These steps are shown in Figure 3.40. Edges of the created formal graphs carry the unique ID as an attribute to establish a link between graph and voxel data. Now, after the algorithm

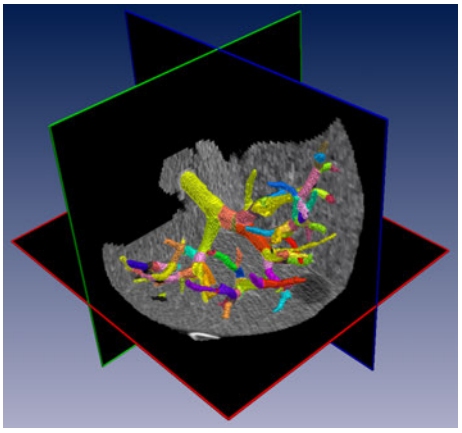


Figure 3.41.: Labeled vessel segmentation where each branch has a unique label visualized with different colors. Image taken from [DOL10a].

has determined edges to be removed, they can also be removed from the skeleton by removing voxels that are labeled with the same ID as the removed edge in the formal graph. Alternatively, new skeletons can be created based on the remaining graph edges. If branch voxels would not get a special label, then trees could fall apart, because branch-voxels are shared between multiple sub-branches. Thus, voxels with a special label are removed only, if there are no other voxels in their 26-neighborhood.

3.6.4.2. Separating vessel segmentations

The separation of interconnected vessel segmentations is done as follows. First, skeletons are labeled with unique IDs as described before. Second, IDs are propagated in the vessel segmentation volume to produce a labeled segmentation. Therefore, the Euclidean distance is used to assign the label of the closest skeleton label to every vessel voxel. The result of this process is shown in Figure 3.41. Now, segmented vessel branches can be removed by deleting voxels labeled the same as the graph edge that was removed by the algorithm. Alternatively, new vessel segmentations can be created based on the remaining graph edges.

3.7. Evaluation framework

Section 2 provided an extensive overview of state of the art vessel extraction techniques. Most of them were validated using either synthetic, simulated or real data by the corresponding authors. Synthetic data usually represent a specific situation that a segmentation algorithm has to deal with. Published results cannot be translated into results with real clinical data. On the other side, real clinical data cannot easily be used to quantitatively evaluate an algorithm. Furthermore, authors are using their own synthetic data or clinical data to which only they have access. This makes a comparison between different algorithms almost impossible. In this section, an evaluation framework is presented that copes with these problems. It allows for a quantitative evaluation and comparison of segmentation algorithms. At its heart, it uses simulated data as ground-truth and a set

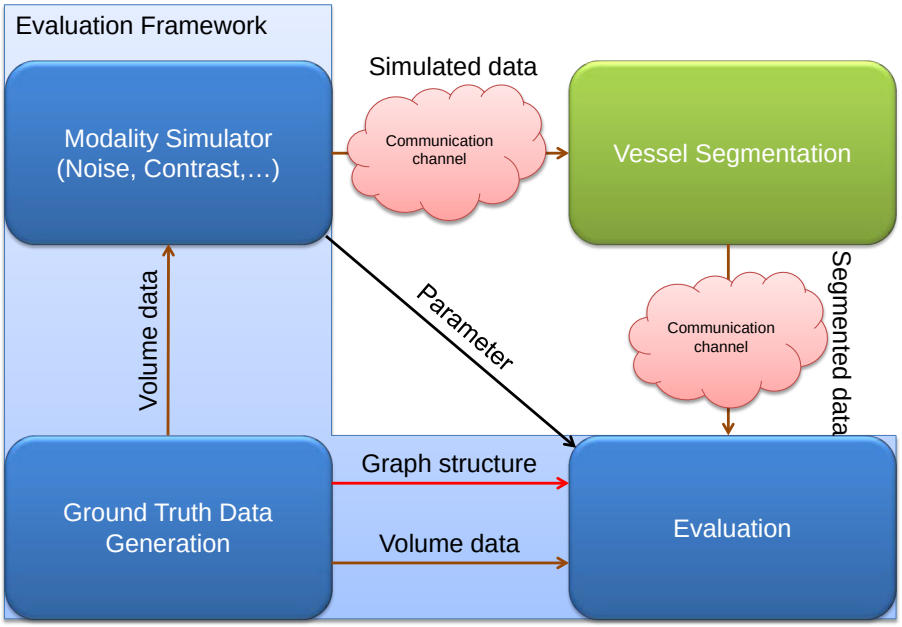


Figure 3.42.: Schematic illustration of the vessel segmentation evaluation framework.

of evaluation metrics that help to analyze various aspects of an algorithm. Using a standardized evaluation framework helps to meet quality criteria like objectiveness, reliability and reproducibility. Although the presented framework was developed with the evaluation of liver vessel segmentation algorithms in mind, it can be extended to other organs, like, for example, the lung.

3.7.1. Overview

The proposed framework consists of three major parts as illustrated in Figure 3.42. At the beginning, ground-truth data has to be created. In this step, volume data and a graph description of the generated vessel tree are produced and serve as the basis for the other two modules. The volume data serves as input to the modality simulator, which perturbs the data with acquisition dependent defects (e.g. noise) and medical related side effects (e.g. increased contrast due to the injection of a contrast agent). The result are simulated datasets with varying contrast and noise that are made available through a communication channel (e.g. on a website, on a memory stick, ...). The receiver of the simulated datasets segments the vessels and returns them through a communication channel to the evaluation framework. In this phase, the evaluation of the received data starts. It uses the ground-truth data (volumetric and graph structure) as reference to calculate quantitative measures. The result of this process are diagrams and tables.

3.7.2. Ground-truth data

The creation of ground-truth data simulates vascular growth into a perfusion volume by iteratively connecting new terminal nodes chosen from some volume constraint by a physical vascular model. Following Karch et al. [KNNS99] and Hamarneh and Jassi [HJ10],

the physical model is defined as follows.

Conservation of flow The flow into a bifurcation point by a parent branch Q_{parent} must equal the flow leaving this point by its left and right children Q_{left} and Q_{right}, respectively.

$$Q_{parent} = Q_{left} + Q_{right}. \tag{3.79}$$

Terminal flow The flow into a terminal node q_{term} must equal the flow through the root node Q_{perf} divided by the number of terminal nodes N, i.e.

$$Q_{term} = Q_{perf}/N. \tag{3.80}$$

Radii relationship The radius of the parent branch r_{parent} is related to its children r_{left} and r_{right} as

$$r_{parent}^{\gamma} = r_{left}^{\gamma} + r_{right}^{\gamma}. \tag{3.81}$$

Hydrodynamic resistance The hydrodynamic resistance R_j of each segment with radius r and length L is given by

$$R_j = \frac{8\eta L_j}{\pi r_j^4}, \tag{3.82}$$

where η is the viscosity of blood. The resistance of a terminal branch $R_{term}^{\star}$ with radius $r = 1$ is given by

$$R_{term}^{\star} = \frac{8\eta L_j}{\pi}. \tag{3.83}$$

Pressure drop Given R_j, the pressure drop ∇P_j along a branch j with flow Q_j is calculated as

$$\nabla P_j = Q_j R_j. \tag{3.84}$$

VascuSynth[4] is a recently published software that is able to simulate volumetric images of vascular trees based on the described physical model. It is based on the paper by Hamarneh and Jassi [HJ10] and available as open source package [JH11]. For this thesis, the software was adapted and integrated into the MITK framework as part of the architecture illustrated in Figure 3.42. Furthermore, the generated formal graphs were extended to match the graphs created in Section 3.5. The functionality of the VascuSynth software is summarized as follows and illustrated in Figure 3.43.

Starting at a user-defined cartesian coordinate within a perfusion volume describing the oxygen demand (called *oxygen demand map* by the authors), the tree generation is performed by iteratively growing a vascular structure while enforcing the physical model described above. During each iteration, a new candidate terminal node is randomly chosen, which will be supplied by an already existing branch. The oxygen demand map is then updated accordingly to reflect the new situation, e.g. it will carry the information that a sub-volume is already supplied with blood. The choice of the existing branch and the exact location of the bifurcation point along this branch is an optimization problem. It is

[4] `http://vascusynth.cs.sfu.ca`

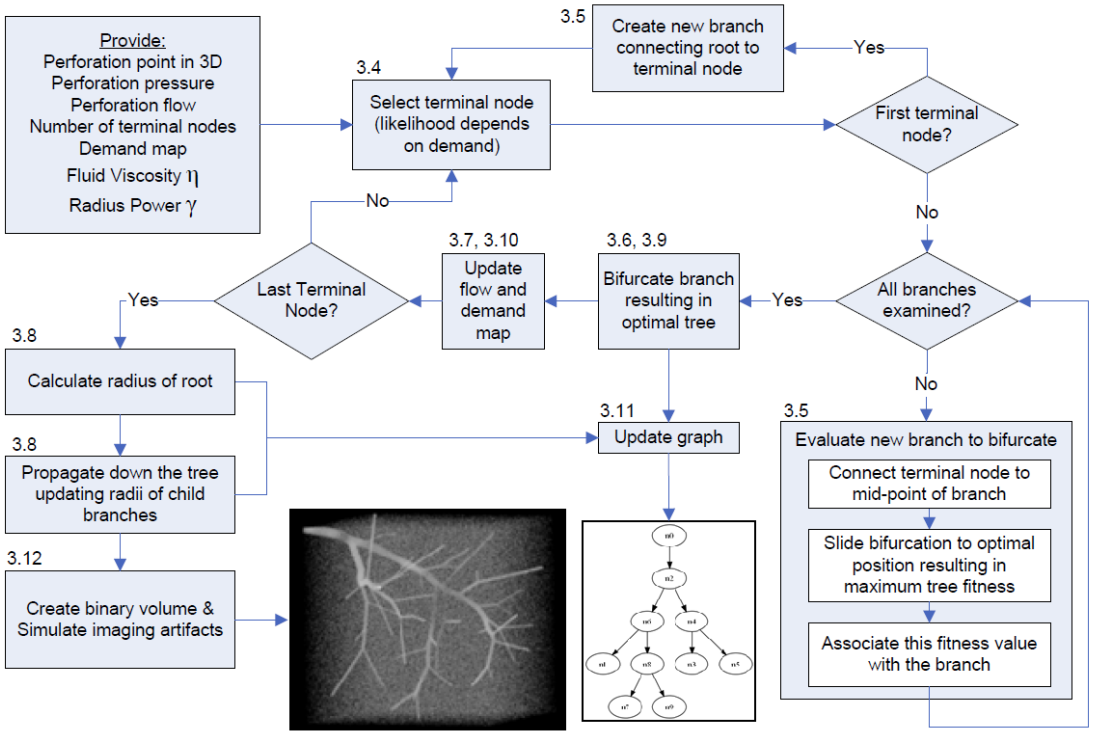


Figure 3.43.: Flow chart of the algorithm implemented by VascuSynth. Image taken from [HJ10].

solved by minimizing the objective function

$$\sum_{\forall j} L_j^{\mu} r_j^{\lambda} \tag{3.85}$$

after connecting the new terminal node temporarily to the center of each existing branch. Increasing λ will favour smaller radii, while increasing μ will favor shorter branches. If an optimal branch is found, the bifurcation point is translated along the branch to further optimize the objective function. Finally, after all terminal nodes have been created, the final radii of the tree branches are calculated according to the physical model.

The algorithm requires as input the position of the root node, the blood pressure at the root node and at terminal nodes, the blood flow at the root node and the number of terminal nodes. Furthermore, it requires an oxygen demand map, which represents the perfusion volume. It is generated using segmented organs from CT datasets. The position of the root node is provided by the user by clicking in the segmented organ. The model parameters for the liver were taken from [KRBWC03] and are summarized in Table 3.4.

Figure 3.44 shows simulated portal veins with a varying amount of terminal nodes. The first row shows a volume rendering of the simulated vessels. The second row shows 2D slices of the simulated vessels overlayed on the used liver mask.

Model parameter	Portal vein	Hepatic vein	Hepatic artery
Blood pressure at root node [mmHg]	15	5	95
Blood pressure at terminal node [mmHg]	8	2	25
Perfusion at root [ml/min]	800	1200	400
Terminal nodes	12.000		

Table 3.4.: Model parameters to simulate hepatic vessel systems.

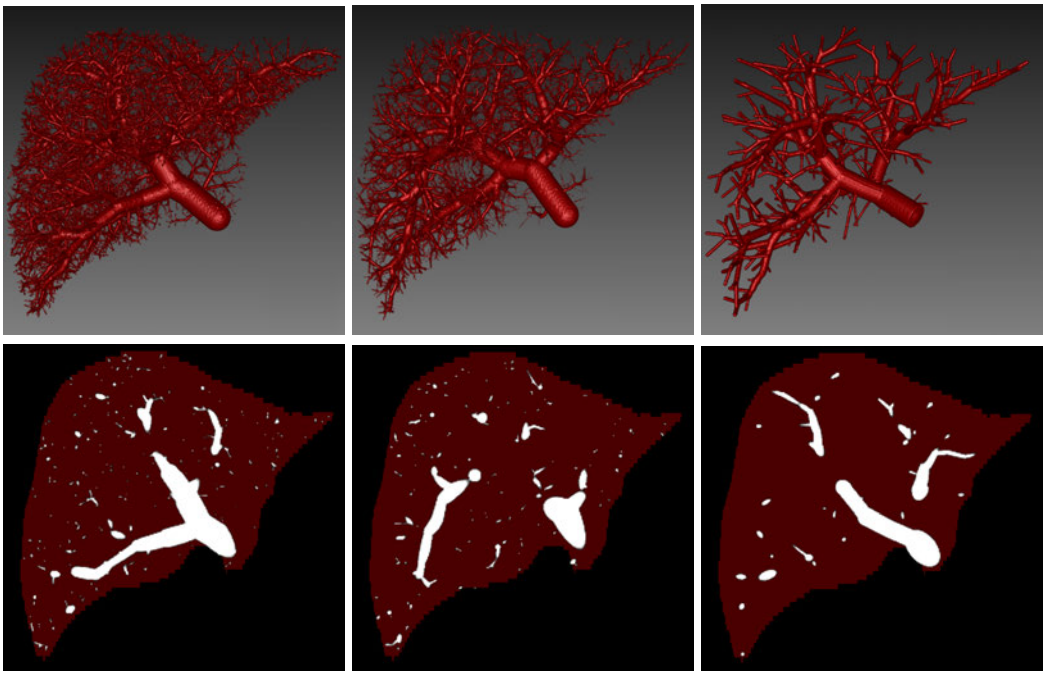


Figure 3.44.: Simulated portal vein. Top row: Volume renderings. Bottom row: 2D slices. From left to right: 12000, 6000 and 500 terminal nodes.

3.7.3. Modality simulation

The image quality that a CT scanner produces is influenced by several components which are

- noise,
- slice thickness,
- reconstruction algorithm,
- radiation dose and
- external contrast enhancement.

Most of them are influenced by technical parameters of the CT scanner, like kVp, mA, exposure time, collimation, reconstruction algorithm, applied filter, table speed and many more. Furthermore, CT scanners from different manufactures behave slightly differently. Adjustment of these parameters usually decrease or increase noise and, simply spoken,

Phase	Arterial	Late-arterial	Portal venous	Late venous
Variable	N_1	N_2	N_3	N_4
Time [s]	25-30	35-40	60-75	90 or longer

Table 3.5.: Time when the contrast agent passes various locations in the human body.

the visibility of the objects of interest. Thus, quantitative evaluation of segmentation algorithms should take components which influence image quality into account. In the current implementation, the framework supports the simulation of contrast enhanced CT datasets where vessel and parenchyma contrast and noise can change. The framework supports the creation of single images with specified parameters or complete dataset series (contrast and noise series) with specified parameter ranges.

3.7.3.1. Contrast

From a medical point of view, a contrast agent is injected intravenously (usually in an arm vein) to enhance vessels and tumors. From there, it then flows into the lungs where it is enriched with oxygen, through the heart and then through the aorta. Near the liver, the aorta has a branch on the underside of the liver, which opens at the porta hepatis in the liver. This is the hepatic artery. After injection, the COA reaches the hepatic artery after N_1 seconds. This is the arterial phase. However, the COA not only flows from the aorta through the hepatic artery into the liver. It also flows through the aorta in the stomach, spleen, etc. From there, the oxygen-poor venous blood flows via the portal vein to the liver. The contrast agent reaches the liver via this route after N_3 seconds. This is the portal venous phase. In both phases, the COA drains through the hepatic vein. However, since only 25% of blood flow is through the hepatic artery and 75% through the portal vein, the COA enhances the hepatic vein much stronger in the portal venous phase than in the arterial phase.

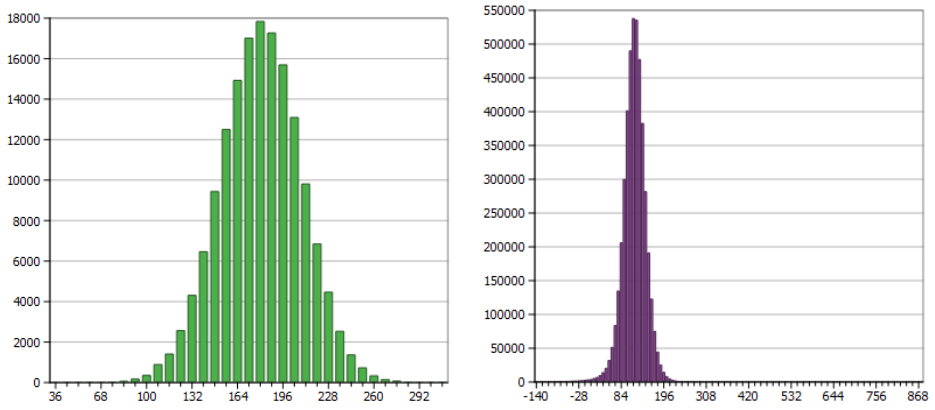
The arterial phase can be further divided into early arterial phase and late arterial phase. In order to detect arterial supplied tumors, the image acquisition is not triggered when the arterial blood flows into the liver, but a moment later (after N_2). It then can be seen if the tumor is enriched with the contrast agent. There is still another phase, the so-called late venous phase (or equilibrium), which is recorded after a total of N_4 seconds. In this phase, the liver is contrasted, but the density decreases in HCCs and thus appears darker on CT.

The times N_i depend mainly on the heart rate, patient size and venous access. After injection of the COA, the CT scanner measures continuously the density in a layer which is located before (in blood flow direction) the region of interest. Once the Hounsfield unit value has risen above a threshold, the recording is triggered. Rough estimates are summarized in Table 3.5.

As a matter of fact, the contrast between parenchyma and vessels changes over time. The CT data shows either vessels with no enhancement, vessels with weak or vessels with strong enhancement with respect to the surrounding parenchyma. In addition, the parenchyma absorbs the contrast agent and is also enhanced over time. The developed evaluation framework allows to create datasets with varying contrast between vessels and

(a) K_m = 0.02 (b) K_m = 0.1 (c) K_m = 0.4

Figure 3.45.: Contrast enhanced simulated datasets.



(a) Histogram for vessels, μ = 172.09 (b) Histogram for parenchyma, μ = 117.75

Figure 3.46.: Histogram for parenchyma and vessels.

parenchyma (contrast series) with constant noise. The contrast is specified as Michelson contrast defined as

$$K_m = \frac{I_{max} - I_{min}}{I_{max} + I_{min}} \tag{3.86}$$

with $K_m \in [0, 1]$.

Figure 3.45 shows examples of contrast enhanced simulated datasets with different Michelson contrasts.

To measure the contrast in a typical clinical dataset, masks for vessels and parenchyma were created using the previously described methods for organ and vessel segmentation. Then, the vessel mask was subtracted from the organ mask. Afterwards, the mean values for parenchyma and vessels were calculated. Figure 3.46 shows histograms for liver parenchyma and liver vessels. The liver vessels have a mean value of μ = 172.09 while the parenchyma has a mean value of μ = 117.75. The Michelson contrast is therefore K_m = 0.18.

	D1.1	D1.2	D1.3	D2	D3.1	D3.2	D4.1	D4.2
Standard dev.	5.7	10.0	8.5	4.4	9.7	10.3	6.7	9.3
Variance	32.8	100.5	71.3	19.3	94.4	106.1	45.0	86.7
Mean	-997.7	-988.5	127.2	-998.8	-991.2	-945.0	-1000.3	-937.8

Table 3.6.: Measured statistical parameters in different datasets.

3.7.3.2. Noise

Noise is a characteristic of all CT scanners which is produced by random fluctuations in the signal. Different types of noise exist, e.g. Gaussian noise, Poisson noise, salt and pepper noise, which occur in different situations. Additive white Gaussian noise (AWGN) is the most common type of noise, which has been integrated into the evaluation framework. Formally,

$$Y = X + Z \tag{3.87}$$

where Y is the observed image obtained by perturbing the original image X with AWGN. Z is an independent and identically distributed random variable which is drawn from a normal distribution with zero mean and variance σ^2

$$Z \sim N(0, \sigma^2). \tag{3.88}$$

To estimate the variance of the normal distribution, different clinical CT datasets were analyzed. Therefore, a homogenous region of size N was extracted. Then the variance was calculated as

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^{N} (x_i - \mu)^2 \tag{3.89}$$

where x_i are the samples in the extracted region and

$$\mu = \frac{1}{N-1} \sum_{i=1}^{N} x_i. \tag{3.90}$$

Figure 3.47 shows noise measurements in homogenous regions in three different CT datasets. For each dataset the used region is shown. Furthermore, plots and histograms are shown, which indicate that the data is normal distributed. Finally, a normal probability plot confirms this assumption.

Table 3.6 summarizes the measured statistical parameters from four datasets and six inspected regions. It can be seen that the variance has a huge variability between datasets and also within datasets, depending where the samples have been taken from. The evaluation framework allows to create datasets with varying noise (noise series) and constant contrast between vessels and parenchyma. Figure 3.48 shows examples of simulated datasets with constant contrast and varying noise.

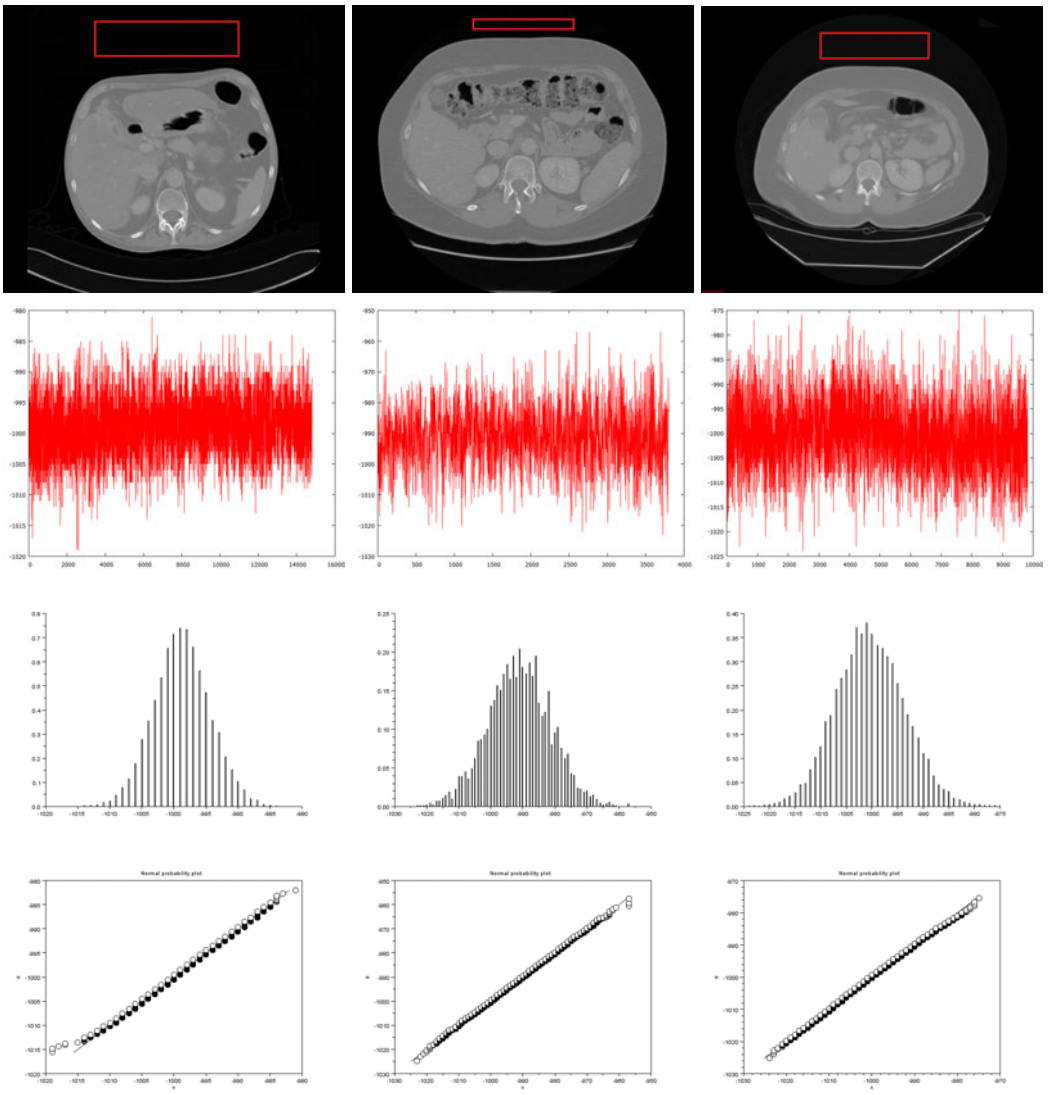
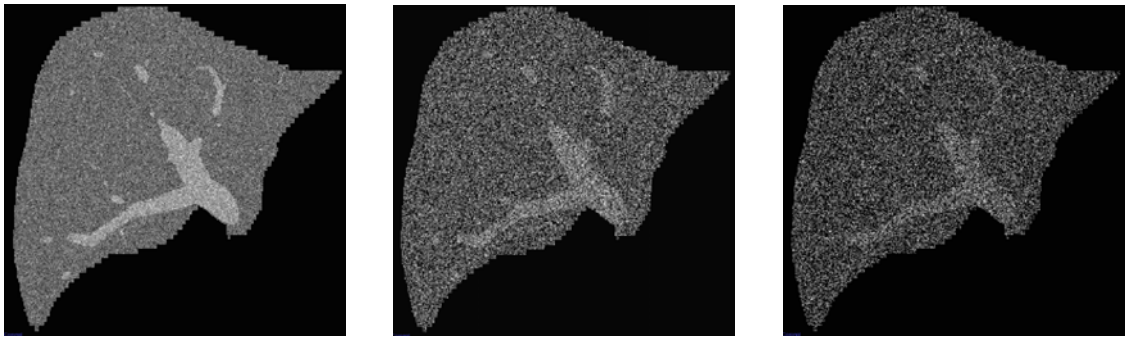


Figure 3.47.: Measurement of the variance in homogenous regions in different datasets. From top to bottom: Screenshots of 2D slices with marked regions that were used for measurements, plot of the measurements, histograms and normal probability plots.



(a) $\sigma^2 = 20$ (b) $\sigma^2 = 50$ (c) $\sigma^2 = 70$

Figure 3.48.: Examples of simulated datasets with constant contrast ($K_m = 0.2$) and varying noise.

3.7.4. Evaluation component

The evaluation component gets as inputs the segmentation results of the generated datasets, the ground-truth data (volumetric image and graph description) and the parameters which were used to generate the simulated datasets. Using these data, image-based metrics can me calculated. To calculate graph-based metrics, formal graphs are created for each segmentation results using the method described in Section 3.5. This is illustrated in Figure 3.49.

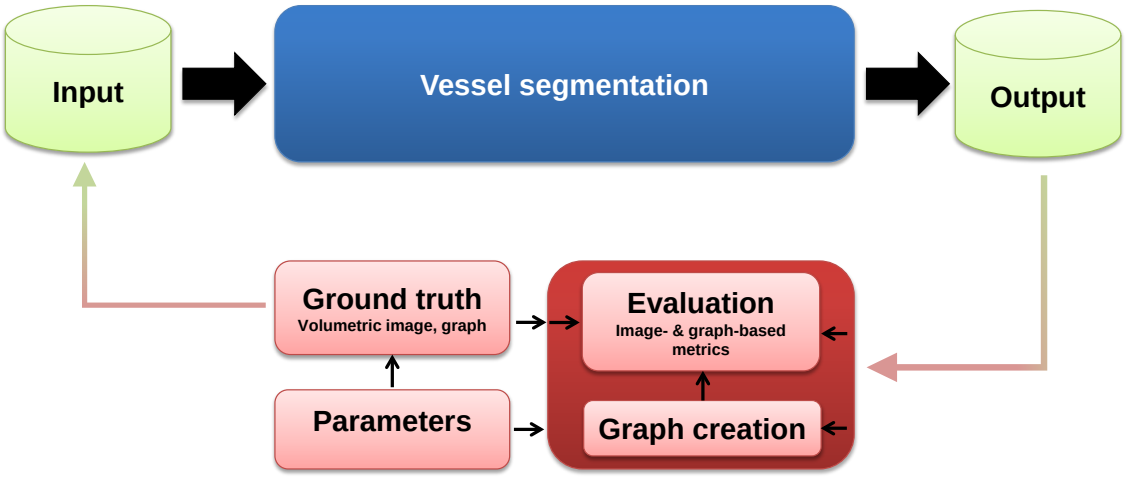


Figure 3.49.: Illustration of the evaluation module.

3.7.4.1. Image-based metrics

Heimann et al. [HTKM04a] recommend to use at least one overlapping metric, the average surface distance and the Hausdorff distance to evaluate segmentation results. This allows to judge different aspects of the segmentation result with respect to the ground-truth data. Overlapping metrics measure how similar two objects are, while distance

metrics can indicate changes in the topology. The evaluation framework implements the following image-based metrics:

- Dice coefficient,
- Hausdorff distance,
- average surface distance, and
- Statistical measures (true/false positives/negatives, sensitivity, specificity).

3.7.4.2. Graph-based metrics

Image-based metrics are quite generic and can be used to evaluate any segmentation algorithm. However, they do not capture domain-specific attributes. Vessels consist of branches with specific length, diameter and volume. Furthermore, vessels have bifurcations and topology. Thus, it makes sense to include additional domain-specific metrics in the evaluation process. Therefore, the evaluation framework implements the following graph-based metrics:

- Number of branches,
- branch length,
- branch volume, and
- branch diameter.

While the number of branches can be visualized in a bar chart, the other attributes require the use of histograms. Here, the bin width is an important factor. Bin widths were calculated according to Scott's normal reference rule, given by

$$h = \frac{3.49\sigma}{n^{\frac{1}{3}}}, \tag{3.91}$$

where σ is the standard deviation of the sample population and n the number of samples. The number of bins is then calculated as

$$k = \lfloor \frac{m_{max}}{h} \rfloor \tag{3.92}$$

with $k \in \mathbb{Z}$.

3.8. Results

In this section, evaluation results of several parts of the proposed vessel extraction methodology are presented.

Clinically, the extraction method was successfully used in an application for intraoperative navigation to prepare CT data and to enhance vessel in Ultrasound data [OLDE*12] and to create formal graphs. Up to date, it was used in 27 patients. Seven cases failed,

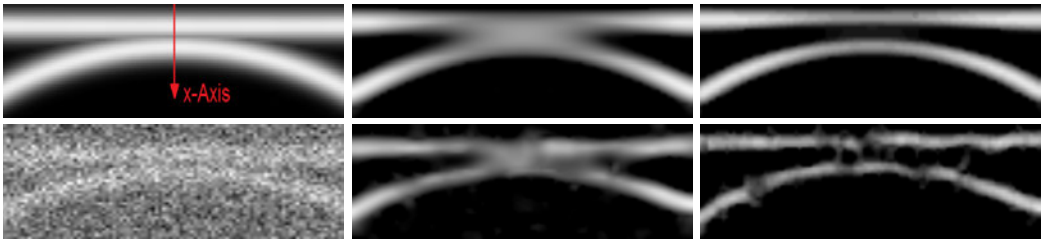


Figure 3.50.: Top left: Original Image. Bottom left: Original Image with added Gaussian noise with σ = 130. Middle: Result using the maximum response approach. Right: Result using the proposed approach with $w_\sigma = \frac{1}{\sigma^\lambda}$. Image taken from [DOL10d].

six of them because of wrong B-Mode settings of the ultrasound device and one because of artifacts in the CT data.

The remainder of this section is organized as follows. Section 3.8.1 compares the proposed weighted additive response to the widely used maximum response approach. Section 3.8.2 evaluates the speed-up of the Laplacian before eigensystem calculations are performed. Section 3.8.3 uses the proposed evaluation framework to evaluate the vessel segmentation algorithm. Section 3.8.4 analyzes the proposed graph creation method and Section 3.8.5 deals with the evaluation of the proposed vessel separation method.

3.8.1. Weighted Additive Response

In the following subsections, some results of the proposed method in comparison to the widely used maximum response approach are shown. Therefore, public available artificial datasets of synthetic vessels were used (Section 3.8.1.1). Gaussian noise with varying standard deviations was added in order to simulate real world conditions.

Furthermore, both methods were applied to the liver and lung obtained from CT datasets (Section 3.8.1.2). In order to reduce the amount of data to be processed, the organs were first segmented from the datasets. For the liver, a model-based segmentation approach was used [EKSW10]. The lung was manually segmented.

The vessels of the liver were segmented using 3D region growing with a manually selected seed point at the root of the portal vein. The thresholds were also chosen manually and, if necessary, iteratively adapted.

3.8.1.1. Artificial datasets

Figure 3.50 shows the results of the maximum response approach (middle) in comparison to results of the proposed method (right) applied to a synthetic dataset showing two very close tubular structures (left). It can be seen that both tubes diffused into one another when the maximum response approach is used. With the proposed method, both tubes are clearly separated. Even in a worst case scenario with added Gaussian noise the proposed approach produced much better results.

It was quantitatively evaluated how good the proposed method is compared to the maximum response approach to separate nearby tubular structures under varying noise

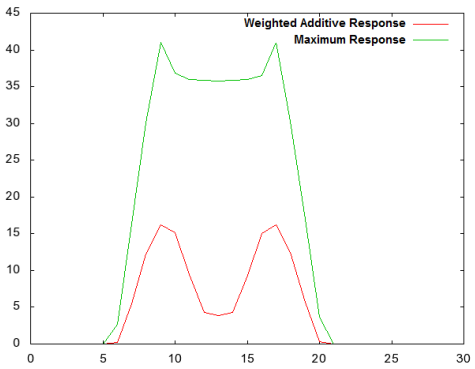


Figure 3.51.: The plotted profiles of the maximum response (green) and the proposed approach (red). It can be seen that with the proposed approach a better separation is achieved. Image taken from [DOL10d].

levels. Therefore, the dataset shown in the top left image of Figure 3.50 was used. Profiles through the tubes in vertical direction for the result using the maximum response approach and the result using the proposed approach were plotted (Figure 3.51). Afterwards, the ratio between the minimum and maximum values (valley-to-hill metric) were calculated, which quantifies how good both tubes are separated. A higher value is better. A value close to one means that it is impossible or at least very hard to segment both tubes. A value below one means that minimum and maximum values are reversed (valley and hill are exchanged). Thus, ideal values should be clearly above one. These measurements were repeated with added gaussian noise with increasing standard deviations up to 300. The minimum and maximum values were always measured at the same position which were determined using the original image. The results are presented in Figure 3.52. In the left graph it can be seen that the valley-to-hill metric for the maximum response approach is in general very close to one, which makes it hard to segment both tubes using thresholding techniques. Starting with a standard deviation of 50 the maximum response approach is too close to one. Thus, both tubes cannot be separated anymore. In the right graph the valley-to-hill metric for the proposed approach is presented. It can be seen that the proposed approach has a much higher buffer to one, which makes it easier to segment both tubes. Up to a standard deviation of 75 it is relatively constant. Between a standard deviation of 75 and approx. 225 the proposed method still outperforms the maximum response approach. The proposed method is more robust against noise and is able to better separate nearby tubes. Qualitative results like the ones in Figure 3.50 confirm these results.

3.8.1.2. Clinical datasets

Figure 3.53 shows the advantage of the proposed method over the maximum response approach when applied to the lung. Nearby vessel structures (e.g. the ones marked with red circles in the first image) are clearly separated with $w_\sigma = \frac{1}{\sigma^\lambda}$. But also with $w_\sigma = 1$, the improvement in comparison to the maximum response approach is visible.

When $w_\sigma = \frac{1}{\sigma^\lambda}$ is used, thick vessels tend to be thinner in comparison to the results

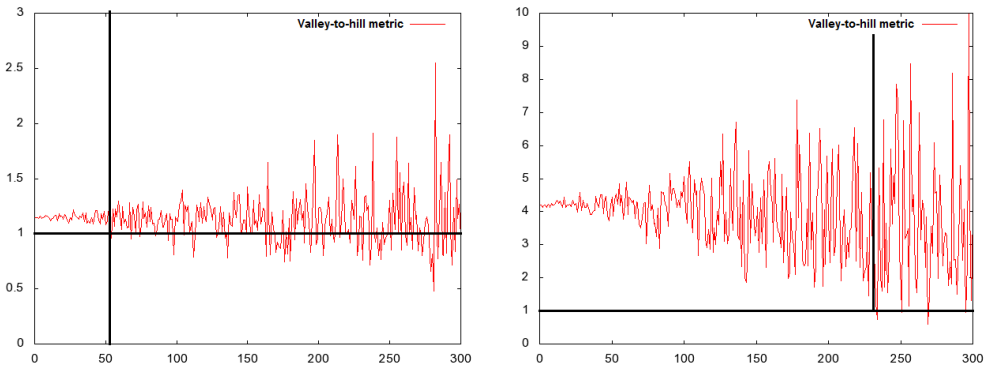


Figure 3.52.: Left: Quantitative results of the maximum response approach. Right: Quantitative results of the proposed approach. Image taken from [DOL10d].

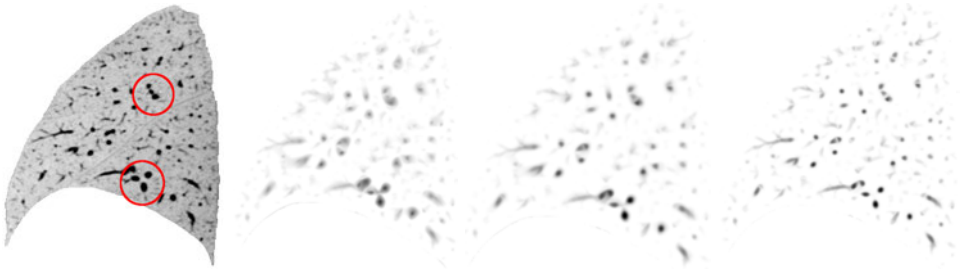


Figure 3.53.: Inverted slices of a CT dataset showing the right lung. Left: Original slice. Middle left: Result using the maximum response approach. Middle right: The proposed approach with $w_\sigma = 1$ Right: The proposed approach with $w_\sigma = \frac{1}{\sigma^\lambda}$. $\sigma = 1.5, 2.0, 3.0$. Image taken from [DOL10d].

of the maximum response approach and to the original dataset. This is because thick vessels are detected at higher scales, which contribute less to the final response than lower scales. However, thinner vessels are more strongly enhanced and nearby vessels are clearly separated. When $w_\sigma = 1$ is used, all scales, including the one containing the maximum response, are added. In this case, the center of a vessel is more strongly enhanced than with the maximum response approach.

Figure 3.54 shows the result of the maximum response approach (left, middle) in comparison to the proposed method (right) applied to the liver. The left image was visualized using linear scalar opacities from 0 for the lowest intensity value to 1.0 for the highest intensity value. The transfer function was adjusted to improve the result. In the middle picture it can be seen that it improved in comparison to the left image. However, smaller vessels are hard to see or are too blurred. The right image was visualized using linear scalar opacities as for the left image. No adjustments were necessary. In comparison to the maximum response approach, it can be seen that the result shows significantly more details and branches are less blurred.

Figure 3.55 shows the segmented vessels of a liver visualized as iso surfaces. With the proposed method it was very easy to find appropriate thresholds. In fact, in most cases fixed values could be used to get good segmentation results. Furthermore, it can be seen that more details were preserved.

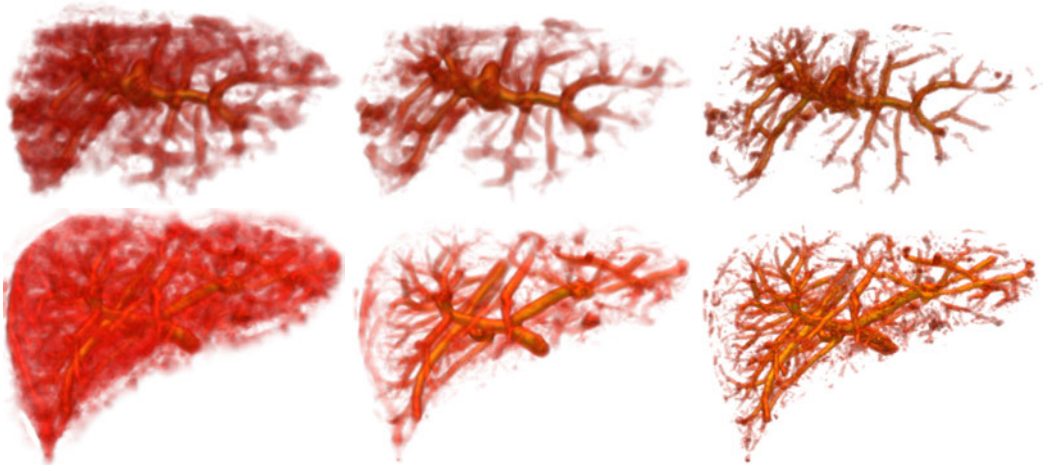


Figure 3.54.: Vessel enhanced volume ray-casting of the liver. Left: Result using the maximum response approach. Middle: Result using the maximum response approach with adjusted opacities. Right: The proposed approach with $w_\sigma = \frac{1}{\sigma^\lambda}$. $\sigma = 2.0, 3.0, 4.0$. Image taken from [DOL10d].

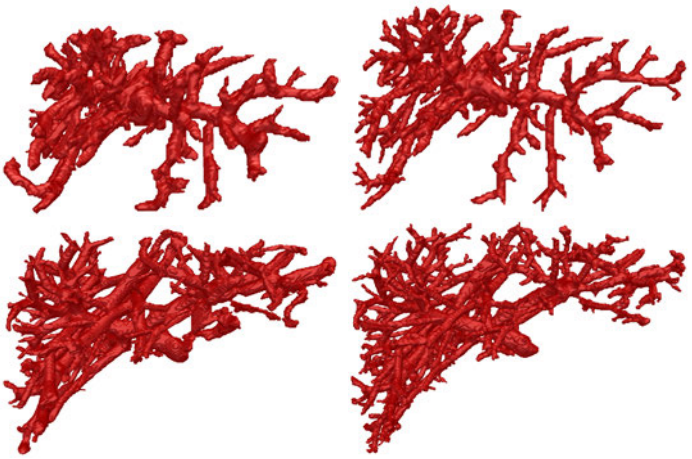


Figure 3.55.: Segmented vessels visualized as iso surfaces. Left: Maximum response approach. Right: The proposed approach with $w_\sigma = \frac{1}{\sigma^\lambda}.\sigma = 2.0, 3.0, 4.0$. Image taken from [DOL10d].

3.8.2. Runtime optimization using the Laplacian

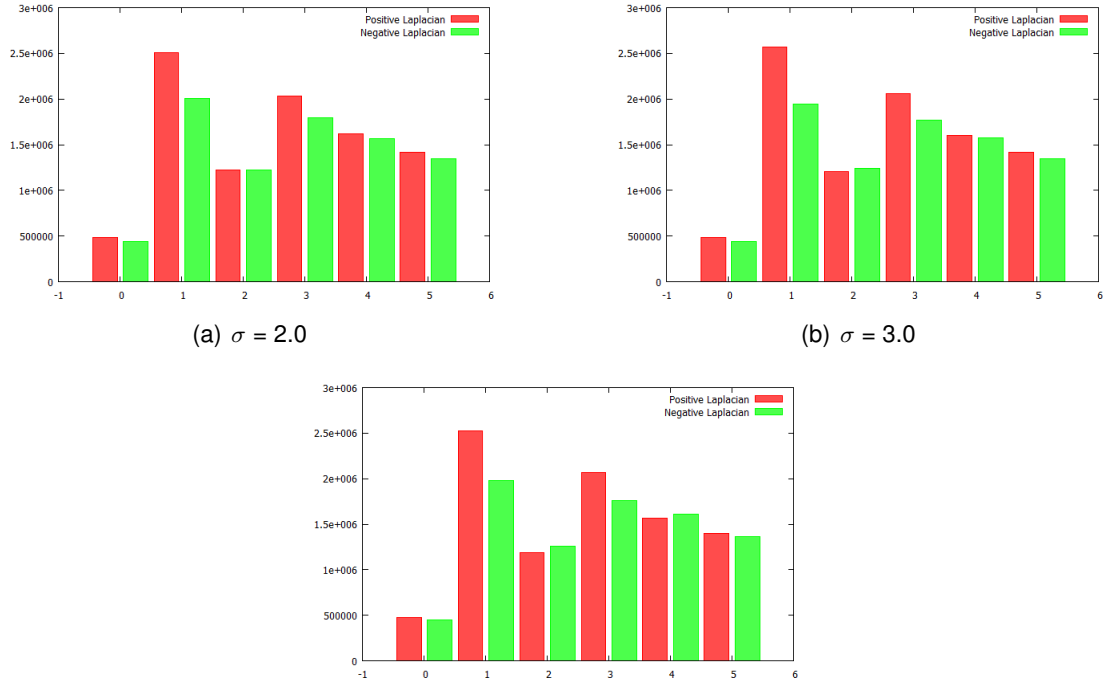
To estimate the computational savings, several datasets were analyzed. From each dataset, the liver was segmented. Then, the Laplacian was calculated for each voxel within the liver and the positive and negative signs counted. The numerical results are presented in Table 3.7. Figure 3.56 shows the results as diagrams.

As described in Section 3.2.4.4, if the Laplacian is positive, the computationally expensive computation of the eigensystem can be skipped. The speed-up at this stage is calculated as

$$Speedup = \frac{\text{Total number of voxels}}{\text{Voxels with corresponding negative Laplacian}}. \tag{3.93}$$

		σ = 2.0		σ = 3.0		σ = 4.0	
Dataset	Total	+	-	+	-	+	-
0	926855	485272	441583	482140	444715	473711	453144
1	4512191	2508618	2003573	2569015	1943176	2529366	1982825
2	2447867	1224520	1223347	1207212	1240655	1189486	1258381
3	3823372	2029306	1794066	2057362	1766010	2063863	1759509
4	3180374	1616711	1563663	1601205	1579169	1569565	1610809
5	2769106	1420087	1349019	1419925	1349181	1402467	1366639

Table 3.7.: Voxels with corresponding positive and negative Laplacian counted at different scales for six datasets.



(a) σ = 2.0

(b) σ = 3.0

(c) σ = 4.0

Figure 3.56.: Voxels with corresponding positive (red) and negative (green) Laplacian counted at different scales (σ = 2.0, 3.0, 4.0) for six datasets. If the Laplacian has a positive sign, computationally expensive computation of the eigensystem can be skipped. It can be seen that this is the case for approximately 50% of the voxels inside the liver.

Table 3.8 summarizes the speed-up across scales for the six datasets. It can be seen that the reached speed-up is almost always in a magnitude order of two, which means that approximately 50% of eigenvalue computations can be saved.

3.8.3. Vessel segmentation

Quantitative evaluation of the vessel segmentation was done using the evaluation framework proposed in Section 3.7. A ground-truth portal vein was created using the parame-

	Speed-up		
Dataset	σ = 2.0	σ = 3.0	σ = 4.0
0	2.09	2.08	2.04
1	2.25	2.32	2.28
2	2.00	1.97	1.95
3	2.13	2.17	2.17
4	2.03	2.01	1.97
5	2.05	2.05	2.03

Table 3.8.: Speed-up of eigensystem calculations across scales for six datasets.

Model parameter	Portal vein
Blood pressure at root node [mmHg]	15
Blood pressure at terminal node [mmHg]	8
Perfusion at root [ml/min]	800
Terminal nodes	500

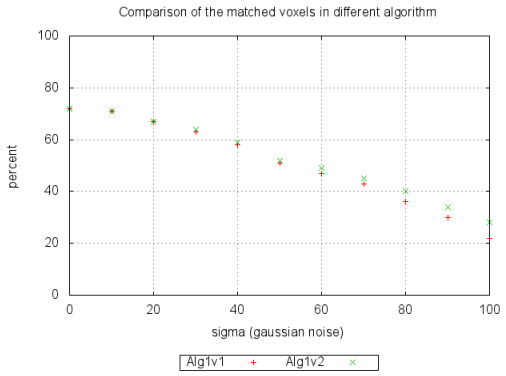
Table 3.9.: Used parameters to simulate the growth of the portal vein.

ters listed in Table 3.9. This was used to create two sets of test datasets. One set was created with varying gaussian noise and constant contrast. The other set with varying contrast and constant noise. Used parameters are listed in Table 3.10. The variance of noise was increased in increments of 10. Contrast was increased in increments of 0.05. Hence, the series of noise datasets consisted of 11 individual datasets, while the contrast series consisted of 8 datasets. Each dataset was segmented manually using the algorithm described in Section 3.3. Therefore, the threshold parameter was adjusted until the result was satisfying. However, because of intra-observer variability, it is expected that results vary between individual datasets. For each dataset two segmentations were created. One (Alg1v1) for the result produced by the segmentation algorithm. And another one (Alg1v2) after applying the volumetric hole closing algorithm described in Section 3.4. As a matter of fact, 22 segmentations were obtained for the noise series and 16 for the contrast series. These results served as input to the evaluation framework, which automatically calculated several metrics and compared both segmentations to each other.

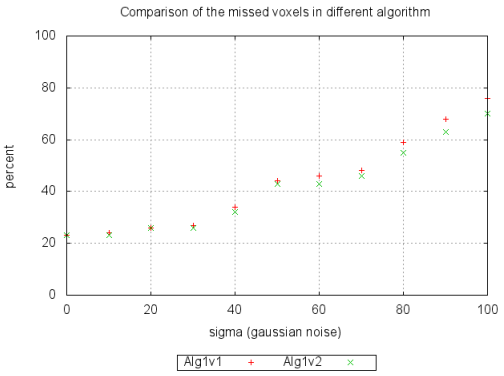
Influence of noise Figure 3.57 shows statistical measures comparing Alg1v1 and Alg1v2 with the ground-truth. In 3.57(a) it can be seen that the number of true positives decreases with an increasing noise variance and that Alg1v2 performs better or equally

	Noise (Variance)	Contrast
Noise series	0-100	0.17
Contrast series	80	0.1-0.45

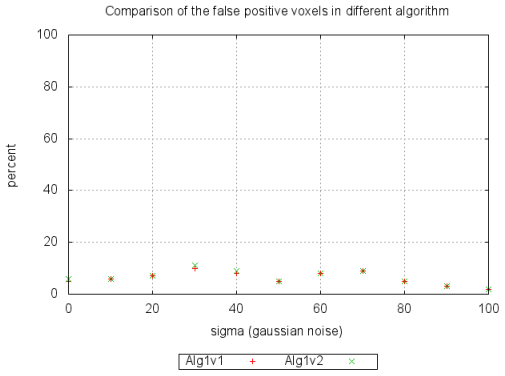
Table 3.10.: Parameters used to create two sets of test datasets to evaluate the effect of varying noise and contrast.



(a) True positives



(b) False negatives



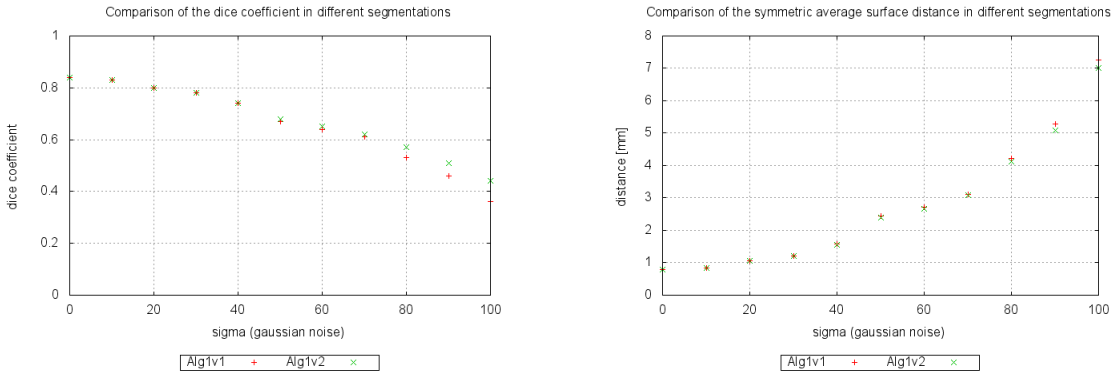
(c) False positives

Figure 3.57.: Comparison of true positives (matched voxels), false negatives (missed voxels) and false positives (wrong voxels).

good. For instance, with a variance of 100 the segmentation of Alg1v2 has approximately 10% more true positives. In 3.57(b), it can be seen that the number of false negatives increases as the noise variance increases. Again, Alg1v2 performs better than Alg1v1. Figure 3.57(c) shows the number of false positives, which is between approximately 5-10%. It can be seen that Alg1v1 and Alg1v2 perform almost similar. Thus, the proposed volumetric hole filling improves the results without increasing wrongly segmented voxels.

Figure 3.58 shows the comparison result using the Dice similarity coefficient as overlapping measure (Figure 3.58(a)) and the symmetric average surface distance as distance measure (Figure 3.58(b)). The former shows that segmentation overlaps decrease with increasing noise variance. The latter shows that the symmetric average surface distance increases with increasing noise variance. In both cases Alg1v2 performs better than Alg1v1.

Influence of contrast Figure 3.59 shows statistical measures comparing Alg1v1 and Alg1v2 with the ground-truth. In 3.59(a) it can be seen that the number of true positives increases with increasing contrast and that Alg1v2 performs better or equally good. In 3.59(b), it can be seen that the number of false negatives decreases as the contrast increases. Again, Alg1v2 performs better than Alg1v1 or equally good. Figure 3.59(c)



(a) Dice (b) Symmetric average surface distance

Figure 3.58.: Comparison of the Dice similarity coefficient and symmetric average surface distance.

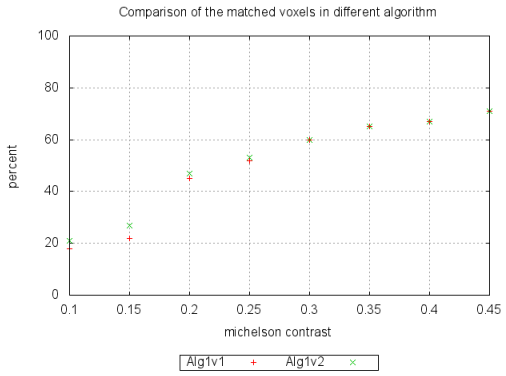
shows the number of false positives. It can be seen that Alg1v1 and Alg1v2 perform almost similar. As in the previous paragraph, it can be noted that the volumetric hole closing did not worsen the result.

Figure 3.60 shows the comparison result using the Dice similarity coefficient as overlapping measure (Figure 3.60(a)) and the symmetric average surface distance as distance measure (3.60(b)). The former shows that segmentation overlaps increase with increasing contrast. The latter shows that the symmetric average surface distance decreases with increasing contrast. In both cases, Alg1v2 performs better or equally good than Alg1v1.

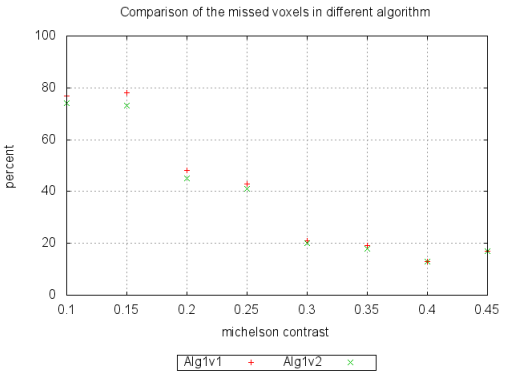
3.8.4. Graph creation

To evaluate the proposed method, the processing chain shown in Figure 3.61 was executed. The segmented portal vein (Figure 3.22) from a contrast enhanced CT dataset of the liver is rotated around one axis in one degree steps. This ensures that most voxel combinations that are possible in 3D will occur. After rotation very thin parts can get disconnected from the rest, thus only the largest connected component was extracted. Holes which can evolve during rotation at the borders of two nearby vessel branches were filled. Afterwards, the proposed methods were applied. To exclude errors in the implementation, regular and end voxels in the classified skeleton were counted and compared with the number of regular and end voxels in the decomposed skeleton. Furthermore, for every voxel in the skeleton it was verified that it is contained in the decomposed skeleton and that no regular or end voxels appeared multiple times in the decomposed skeleton. This helped a lot to find problems that can occur during skeleton traversal. Finally, no abnormalities were found anymore.

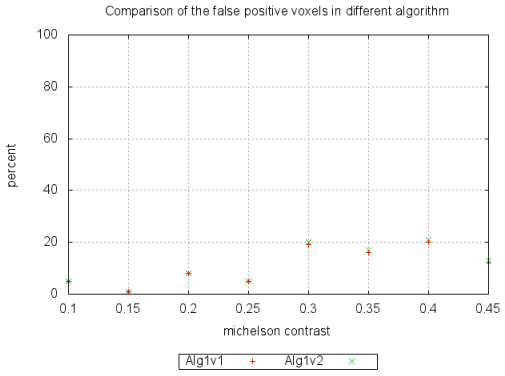
Figure 3.62(a) shows the number of sub-branches and branch voxels in the decomposed skeleton after applying Equation 3.37 and Equation 3.38. It can be seen that both measures correlate, and that they are not invariant under rotation. This is important for the development of tree matching algorithms. They must be able to handle cases where two trees are extracted from one subject at different times or through different modalities.



(a) True positives



(b) False negatives

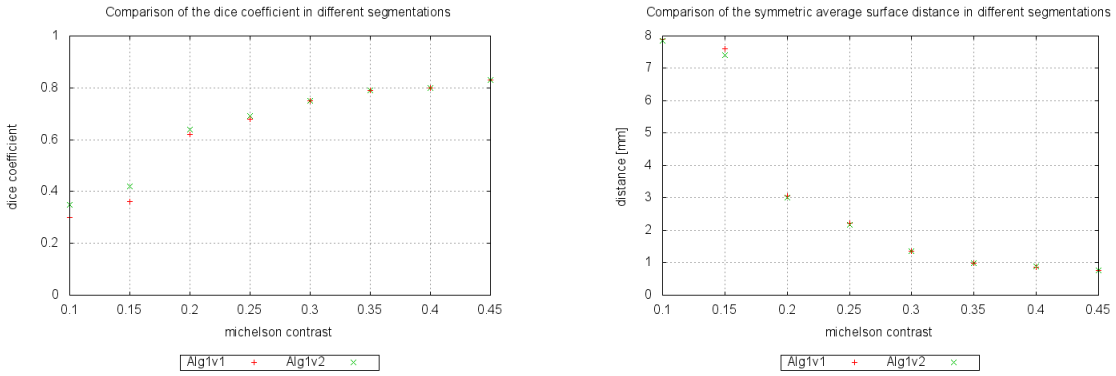


(c) False positives

Figure 3.59.: Comparison of true positives (matched voxels), false negatives (missed voxels) and false positives (wrong voxels).

Even if both trees visually look the same, they will differ in the number of branch nodes and branches. This is mainly due to the fact that two segmentations will not be exactly equal if the volumes were acquired under different conditions. This in turn will result in different skeletons, which leads to a different number of branch voxels and sub-branches. In Figure 3.62(b) the distribution of sub-branches is shown. It follows almost a normal distribution as can be seen in the normal probability plot in Figure 3.62(c). Only at the upper end a deviation from normality (straight line in the figure) is visible. This insight can be used to compare and classify different graph creation methods. An ideal method (from skeletonization to graph creation) would produce a constant number of sub-branches, even under rotation. Many additional sub-branches appear during rotation, because of two neighboring branch voxels that are treated as a sub-branch, or because additional skeleton branches are created by the skeletonization algorithm. The former cause can probably be eliminated by clustering branch voxels into junctions [Kle06]. The latter by studying and improving the skeletonization algorithm.

Figure 3.63 shows a visualization of the skeleton. It was very easily generated using the decomposed skeleton just by connecting neighboring voxels within a sub-branch as polylines.



(a) Dice (b) Symmetric average surface distance

Figure 3.60.: Comparison of the Dice similarity coefficient and symmetric average surface distance.

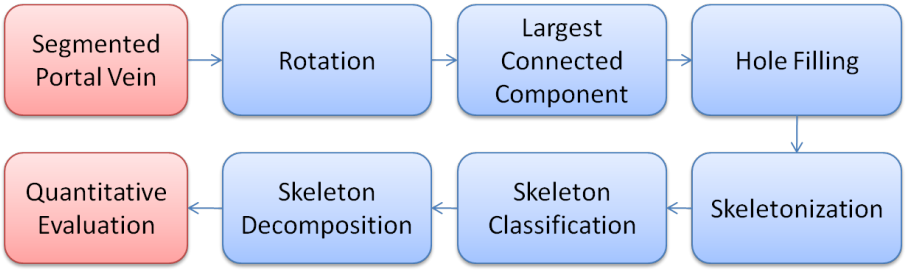


Figure 3.61.: Processing chain used during evaluation. Image taken from [DOL10c].

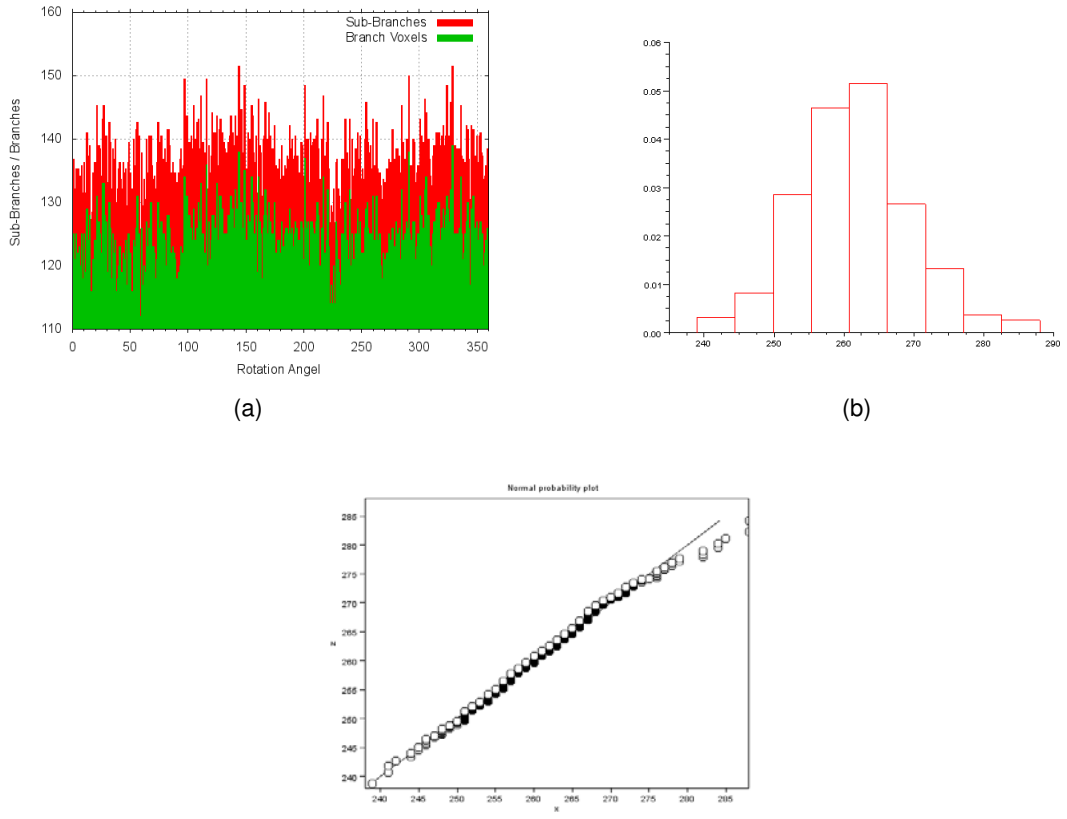
3.8.5. Vessel separation

The proposed algorithm depends on some thresholds. They were experimentally determined and set to fixed values for all experiments as follows. $S_{path} = 0.5, S_{curv} = 1.75, S_{angle} = 1.5, S_{diam} = 0.8, S_{pred} = 1, S_{suc} = 4, S_{pl} = 40$. The majority voting was applied with parameters $r = 1$, $m = 1$ and $m = 1$. All experiments were executed on a PC equipped with an Intel Core I7 CPU 2.93 GHz, 8 GB RAM and Windows 7 x64 as operating system.

Figure 3.64 shows the proposed separation algorithm applied to one dataset. First, root nodes are manually selected (Figure 3.64 top left). Second, erroneous edges are detected and automatically highlighted (Figure 3.64 top right). Third, additional edges to be removed can be manually marked (Figure 3.64 bottom left). Fourth, with this information the liver and portal vein can be separated (Figure 3.64 bottom right).

Figure 3.65 shows how the implemented application assists the user in the case of a fully manual separation. Each path between roots of two vessel trees is sequentially visualized and the user can concentrate on selecting the wrong edge on that path. This technique is also used in case the automatic separation fails for some edges.

Figure 3.66 shows the result after separating the vessel segmentations using the results from the graph-based separation algorithm. Both vessel trees are visualized in colors medical experts are familiar with, namely blue for the liver vein and red for the portal



(a) (b)

(c)

Figure 3.62.: (a) Number of branch voxels (green) and sub-branches (red) during rotation. The amount of sub-branches were divided by 1.9 for visualization purposes. (b) Distribution of sub-branches. (c) Normal probability plot. Image taken from [DOL10c].

vein.

A quantitative evaluation of the proposed separation algorithm was also performed. Therefore, nine interconnected graphs of the liver vasculature generated from CT datasets of the liver were used. Wrong edges were searched manually and thus were known before applying the algorithm. The results are presented in Table 3.11. On average a dataset contained 275.1 edges with 4.9 interconnections and 2.5 loops. 4.2 interconnections (85.7%) and 2.2 loops (88%) were successfully detected. In datasets eight and nine, the detected interconnections were wrong. However, a manual correction was easily possible. On average the algorithm finished after 0.11 secs, but even in the worst case a result was achieved in approx. 0.5 secs (interactive corrections are not included in this measurement). Figure 3.67 shows some of the wrongly detected edges. False detections were due to ambiguous situations. No matter from which side the path is traversed, model assumptions are met or violated, so that identification of the interconnecting edge was not possible.

Figure 3.63.: Visualization of a skeleton as polyline easily created using the proposed decomposition scheme. Image taken from [DOL10c].

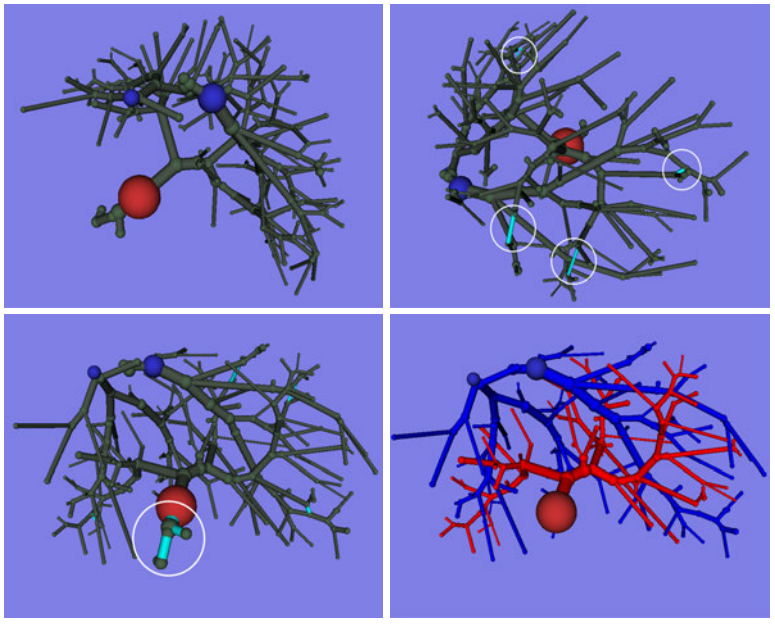


Figure 3.64.: Single steps of the proposed algorithm to separate interconnected vessel trees. Top left: User selected root nodes of portal vein (red) and liver vein (blue). Top right: Wrong edges determined by the proposed algorithm are highlighted (turquois). Bottom left: User can mark additional edges that should be removed. Bottom right: Result after removing all marked edges. Image taken from [DOOL11].

3.9. Discussion

A methodology for the extraction of hepatic veins was presented. From an algorithmic point of view it consists of several processing steps. After organ segmentation, vessels are enhanced using a multiscale Hessian-based analysis. Therefore, a novel multiscale integration approach, called *Weighted Additive Response*, was presented. The proposed

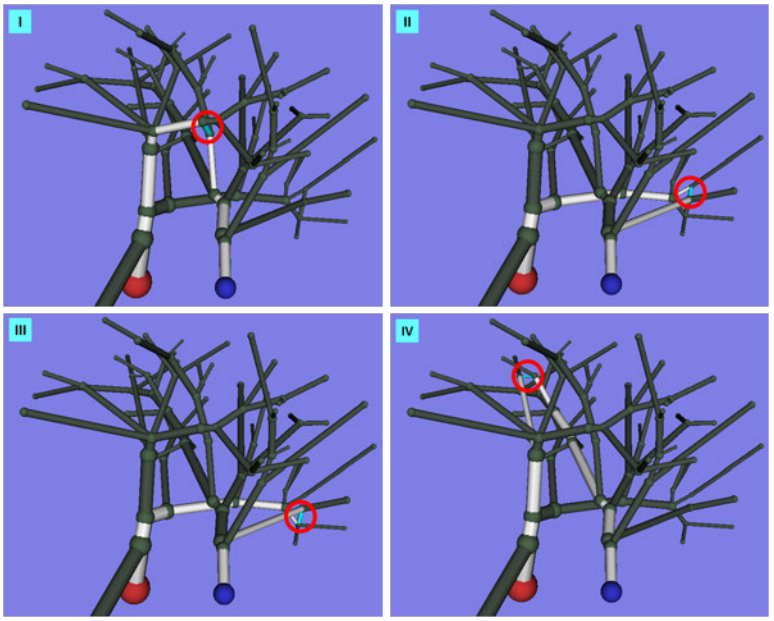


Figure 3.65.: Visualization of paths between roots of different vessel systems and manually selected interconnections. Image taken from [DOOL11].

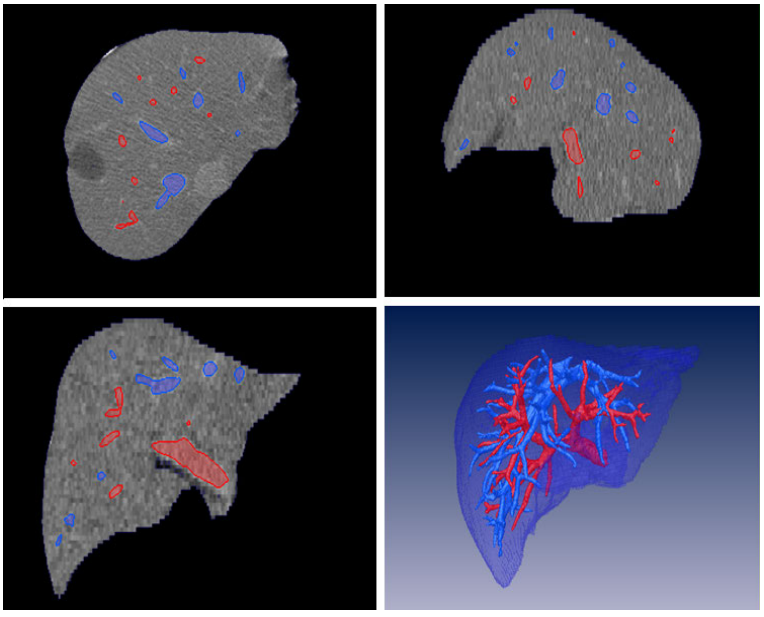


Figure 3.66.: The result after applying the proposed processing chain to a CT dataset of the liver. Top left, right and bottom left: Transversal, Sagittal and Coronal slices of the segmented organ with portal vein overlayed in red and liver vein in blue. Bottom right: 3D visualization of the result. Image taken from [DOOL11].

method was applied to synthetic vessels and to the liver and lung from clinical datasets. It was compared to the widely used maximum response approach and quantitative and qualitative results were presented. The results showed that with the proposed approach the diffusion of nearby tubular structures into one another is significantly reduced. Fur-

Dataset	Edges	IC/LP	TP	FP	RT
1	141	5/-	5/-	0/-	0.07
2	249	5/-	5/-	0/-	0.2
3	92	1/1	1/1	0/0	0.02
4	372	9/7	9/6	0/1	0.5
5	254	1/4	1/4	0/0	0.02
6	455	3/4	3/2	0/2	0.04
7	242	3/0	3/0	0/0	0.02
8	214	3/1	2/1	1/0	0.01
9	457	14/6	9/6	5/0	0.11
Sum	2476	44/23	38/20	6/3	0.99
Avg	275.1	4.9/2.5	4.2/2.2	0.7/0.3	0.11

Table 3.11.: Quantitative evaluation results of the separation algorithm. IC: Interconnections, LP: Loops, TP: True Positives, FP: False Positives, RT: Run Time in secs. Table partly taken from [DOOL11].

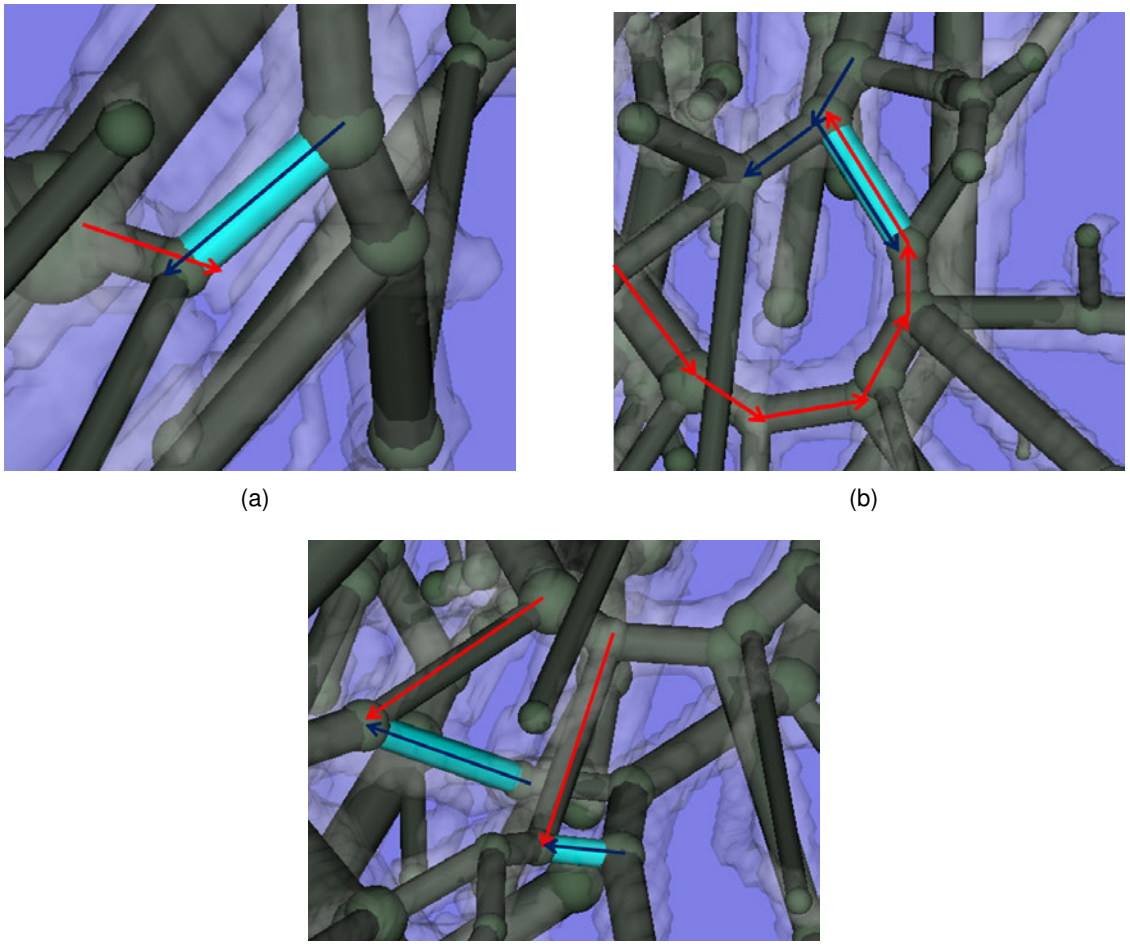
	Maximum response	Weighted additive response
Junction coherence	+	+
Nearby vessels separation	-	+
Separation from background	-	O
Robustness against noise	-	+

Table 3.12.: Rating of multiscale integration approaches. The proposed weighted additive response has several advantages compared to the maximum response approach.

thermore, tubular structures are more strongly enhanced towards their centers, making segmentation using e.g. region growing much easier. Both properties also greatly improved the quality of ray-casted vessel enhanced volume renderings. The presented approach is also more robust against noise than the maximum response approach. Table 3.12 summarizes the properties of the weighted additive response approach compared to the maximum response approach.

The proposed method could have an impact in the sense that almost every published work so far that uses the maximum response approach could switch with minimal effort to the proposed method to significantly improve their results. It is not limited to a specific vesselness function, but can be used in combination with other filter functions, too. In preliminary experiments, similar results as the ones presented were achieved when using e.g. the filter function presented by Frangi et al. Therefore, it can be concluded that the proposed *Weighted Additive Response* approach is an improvement over the widely used maximum response in terms of separation of nearby vessels and robustness against noise.

Based on the idea of the divergence of the gradient vector field, a novel vesselness function was proposed. The divergence of the gradient vector field results in the Laplacian of Gaussian when embedded in a multiscale analysis framework. It was shown that the results are equal to the results produced by Erdt et al.'s filter function. The Laplacian



(a) (b) (c)

Figure 3.67.: Examples of wrongly detected edges in dataset 8 and 9. They are due to ambiguous situations. If the path from root to root is traversed in both directions, model assumptions are met or violated.

of Gaussian can be further utilized to reduce the calculation of eigenvalues and eigenvectors. Typically, this is an expensive operation as it involves the solution of an equation system. Experiments have shown that these calculations could be reduced by approximately 50%.

Vessel-enhancement filters are usually based on a tubular model. This model assumption is violated at ramifications. At these locations, the filter response is low. Depending on the used segmentation algorithm, this results in holes or tunnels. A filter based on majority voting produced good results. However, it was shown that one filter is not enough. Either big holes and tunnels are not filled or small holes are left over. To correct these defects, a cascaded volumetric hole closing algorithm was proposed. It is well suited to deal with holes introduced by vesselness filters at ramifications without altering the overall shape of the vessels.

For further processing of the extracted vessels, a formal graph structure has to be obtained. Therefore, the hierarchical decomposition of vessel skeletons was presented. The proposed method simplifies the process of formal graph creation and feature extraction.

To achieve this goal, a measure for proper branch voxel classification was proposed and a state-machine for voxel traversion was introduced. It was shown that using the proposed decomposition scheme feature extraction and visualization can be easily done. Furthermore, it was found that the number of sub-branches follows a normal distribution under rotation, which has implications for tree matching algorithms.

Due to low resolutions and partial volume effects, hepatic veins can be interconnected at some points. Therefore an algorithm to separate interconnected hepatic veins was proposed. It takes the differences between the vessel structures in a real physical liver and the mapping of these structures using imaging algorithms into a discrete volume into account. Qualitative evaluation has shown that the proposed method is viable and quantitative evaluation has confirmed that most interconnection edges and loops were correctly detected. A manual correction was easily possible in case of misclassification. With an average runtime of approx. 160 ms the proposed method to separate interconnected hepatic veins is very fast. In the worst case, a graph with 372 edges could be separated in approx. 500 ms. Separation results are transfered back to the imaging data, which can be used to overlay the separated vessels in different colors on the original CT data.

A disadvantage of current vessel segmentation evaluation methodologies is that they are not standardized. Public available datasets are limited to simple artificial data. Public available clinical datasets usually do not come with manual segmentations. And even if, then they are subject to inter- and intraobserver variability. Thus, results published by different research groups are therefore not objectively comparable. A contribution of this thesis was therefore a standardized evaluation framework. It simulates vessels based on physiological principles and adds defects like noise and varying contrast, caused by the scan process, to this data. Provided segmentations are fully automatically evaluated and corresponding diagrams and graphs generated. With some extensions it could be used to setup a web-based vessel segmentation challenge in the future. In this thesis, it was used to evaluate the segmentation results created using a combination of a wavefront propagation technique and the proposed vessel enhancement method.

4. Interventional planning

Parts of this chapter were previously published by the author in the following conference papers:

- DRECHSLER K., STROSCHE M., OYARZUN LAURA C.: Automatic roi identification for fast liver tumor segmentation using graph-cuts. In *Proceedings of SPIE Medical Imaging* (2011)
- DRECHSLER K., OYARZUN LAURA C.: Simulation of portal vein clamping and the impact of safety margins for liver tumor resections. In *Abdominal Imaging. Computational and Clinical Applications* (2012), pp. 51–59
- DRECHSLER K., OYARZUN LAURA CRISTINA ANS WESARG S.: Interventional planning of liver resections: An overview. In *Proceedings of the 34th Annual Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)* (2012), pp. 3744–3747
- DRECHSLER K., ERDT M., OYARZUN LAURA C., WESARG S.: Multiphase risk assessment of atypical liver resections. In *Proceedings of the 25th IEEE International Symposium on Computer-Based Medical Systems (CBMS)* (2012), pp. 1–4

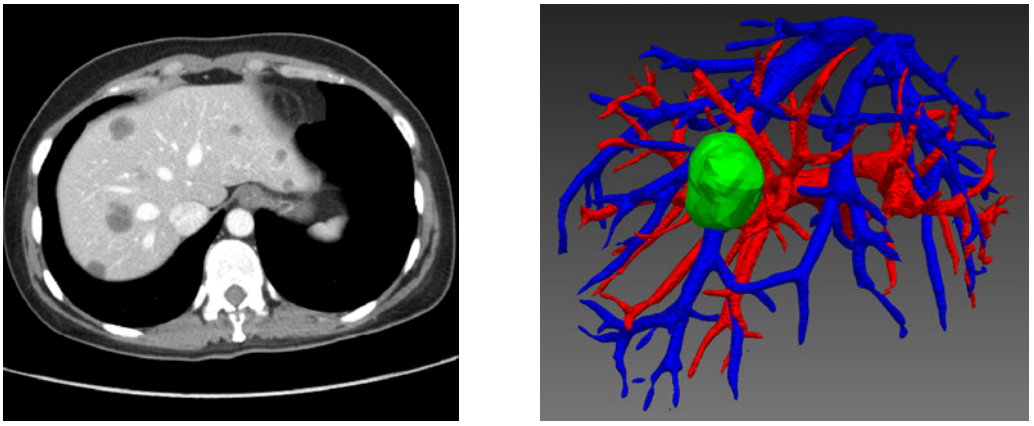
4.1. Introduction

Liver cancer is the third most common type of cancer. Among available treatment options, a surgical resection offers the best prognosis for long-term survival. It is therefore important that such a surgical procedure is carefully prepared. Modern computer technology offers convenient ways to simulate different resection scenarios and helps to determine the best treatment for a given case. Computer-based planning systems were proposed to support surgeons to prepare optimally for an intervention.

In this chapter, a planning system for liver resections is developed that utilizes the vessel extraction methodology described in Chapter 3. This chapter starts with a review of current state of the art planning systems (Section 4.2). They are reviewed according to their medical use case, e.g. if they support typical (Section 4.2.1) or atypical (Section 4.2.2) resections. A table summarizes these information together with a classification of used methods according to the classification scheme presented in Chapter 2.

Section 4.3 provides an overview of the developed system, which consists of several modules. It is the first application that utilizes deformable registration to fuse information from several datasets.

In Section 4.4 a module for interactive tumor segmentation based on graph cuts is presented. Section 4.5.1 describes a module for typical resections. A module to assess



(a) 2D visualization (b) 3D visualization

Figure 4.1.: Visualization of hepatic veins and the tumor in 2D and 3D. 3D visualization helps to orient oneself and learn about the relationships between these major structures. Image (b) taken from [DOL12b].

the risk of atypical resections is presented in Section 4.5.2. In Section 4.7 the developed system is discussed.

4.2. State of the art

Computer-based planning systems provide tools to support the surgeon during his decision making process and to plan treatment of a patient. One feature that every planning system provides is 3D visualization. This helps the surgeon to orient oneself to get an idea where the tumor is located and how the main hepatic vessels are spread within the organ. Leeuwen et al. [vLNHF95] integrated 3D visualization in a very early planning system. Beermann et al. [BTB*10] showed that 3D visualization significantly improves understanding of surgical liver anatomy. Figure 4.1 shows a side-by-side comparison of 2D and 3D visualization. The 2D views show a contrast enhanced CT dataset. Vessels are brighter than the surrounding parenchyma, while the tumor appears darker. The 3D view shows both main vessels systems (portal and hepatic veins) and a tumor in different colors. The relationship between these structures are clearer in the 3D view.

In the following sections, computer-based planning systems for liver tumor resections are reviewed according to their medical use-case (e.g. typical or atypical resections).

4.2.1. Typical anatomic resections

For surgical practice, the liver is divided into several functional independent segments. Each segment has its own arterial and portal venous supply and hepatic venous and biliary drain. Claude Couinaud suggested to divide the liver into eight segments by the third order branch of the portal vein. Later, Bismuth proposed three planes aligned along the three main hepatic branches and the portal vein to classify the liver. Jean H.D. Fasel [Fas08,FMP10] noted that due to anatomical variations, Couinaud's classification scheme is questionable. He investigated the branching patterns of the liver and found that the

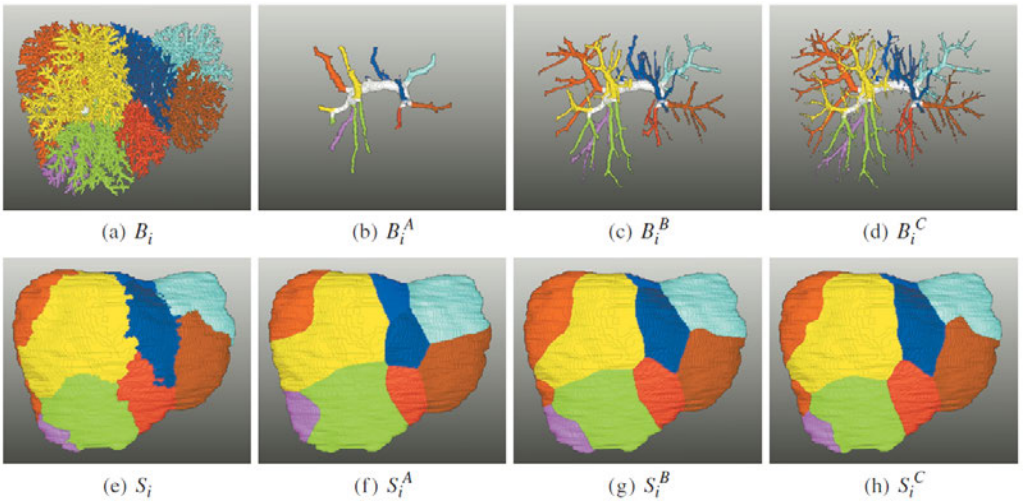


Figure 4.2.: Evaluation of the nearest neighbor segment approximation approach using corrosion casts of the liver by Selle et al. [SSPP00]. The portal venous vessel system is extracted from a CT scan of a corrosion cast of the liver (a) and pruned with varying degrees (b-d). Corresponding Couinaud segmentations (f-h) are compared with the ground-truth segmentation (e). Image taken from [SSPP00].

number of second order branches was between 9 and 44. This indicates that the liver does not consist of eight segments, as proposed by Claude Couinaud, but of many more. He proposed a flexible 1-2-20 scheme to classify the liver. A review of different liver classification schemes was published by Rutkauskas et al. [RGP*06].

During an intervention, the surgeon can clamp the portal vein at a position that supplies blood to the (sub-)segment where the tumor is located. This leads to a color change on the surface of the liver, which indicates the borders of this (sub-)segment. The surgeon then usually uses a coagulator to 'draw' the borders on the liver surface before releasing the clamped vein.

A big advantage of anatomical resections is that no main vessels are close to segmental boundaries. Bleeding is therefore reduced by completely resecting one or more segments by cutting along these boundaries. Computer-based planning systems for anatomic resections allow for the calculation of liver segments. They can be classified into vessel-based methods and plane-based methods.

Glombitza et al. [GLD*99] presented a planning system that provides vessel and plane based tools to visualize liver segments for typical resections.

Selle et al. [SSPP00] proposed to classify a segmented liver using annotated portal veins. Therefore, they segment the liver vessels using a region-growing-based method. The segmentation usually contains both, the portal and hepatic veins. They are interconnected at some points due to partial volume effects and false positive segmentation results. To separate them, they transform the vessels into a formal graph which is analyzed for violations to some model assumptions. The graph is also used to label branches of the portal vein. This is then used to calculate liver segments as follows. Let L be the set of liver voxels and P_j a portal vein branch with label j, $j = 0, .., n-1$. The function $g(x, y)$

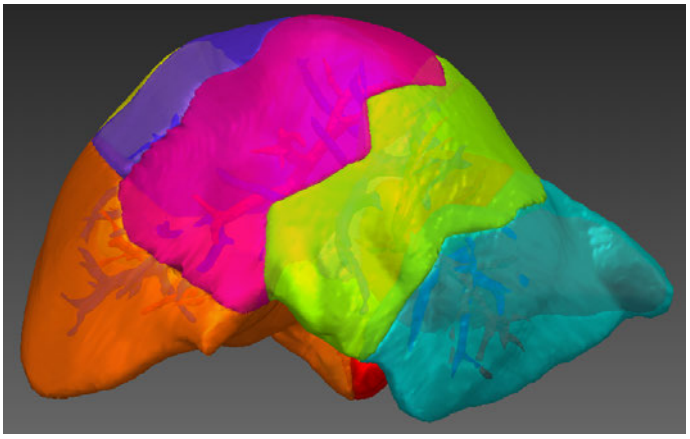


Figure 4.3.: Patient-specific liver segments visualized as semi-transparent surfaces. The hepatic veins can be seen through the surface in red (portal vein) and blue (hepatic vein). Image taken from [DOL12b].

assigns label y to voxel x and $dist(x, y)$ calculates the Euclidean distance between voxels x and y. Then

$$\forall v \in L.g(v, f(v)), \tag{4.1}$$

with

$$f(v) = \underset{j=0,\ldots,n-1}{\operatorname{argmin}} \; \underset{v_i \in B_j}{\operatorname{argmin}} \, dist(v, v_i). \tag{4.2}$$

The result is an approximation of the portal venous territories with respect to the defined labels. The accuracy of this method was evaluated using corrosion casts (Figure 4.2) of the liver and found to be between 80-90%. Later work built on these results [HPSP01, SPSP02, FCN*04, KVPL04, HZH*09, SHP11]. Figure 4.3 shows patient-specific liver segments calculated using the approach proposed by Selle et al. [SSPP00].

Soler et al. [SDM*01] proposed to calculate the portal venous territories for all branches. These branches are then merged in a bottom-up manner using anatomical information from a labeled atlas that is registered to the current dataset.

In the web-based system developed by Meinzer et al. [MSS*04], different CT phases can be registered to combine vessels visible in each phase using an affine approach. After interactive separation of interconnected vessels, dependent tissue is calculated. Resection lines are visualized on the surface of the liver and volumetric analysis can be performed. The system allows access to the generated data over a secure virtual private network (VPN) connection.

Reitinger et al. [RBBS06] developed an augmented reality system with stereoscopic 3D visualization and 3D interaction for manipulating a liver model. They used the methodology proposed by Beichel et al. [BPJ*04] to segment liver vessels and transform them into a formal graph representation. Liver segments are calculated using the approach by Selle et al. [SSPP00]. For anatomical resections, the user can select individual segments for removal and perform quantitative analysis. Figure 4.4 shows a typical resection using a tracked pencil to select and remove complete segments. An evaluation of this system

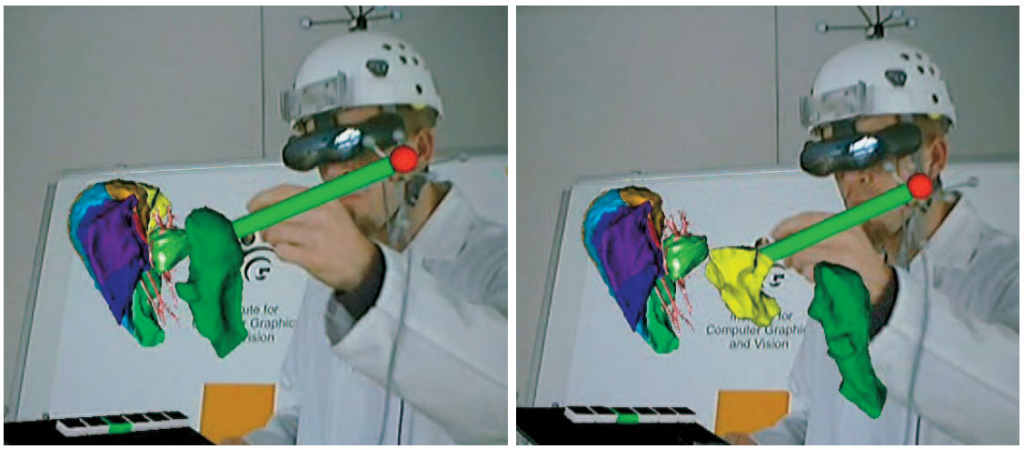


Figure 4.4.: An augmented reality system for planning liver resections. Image taken from [RBBS06].

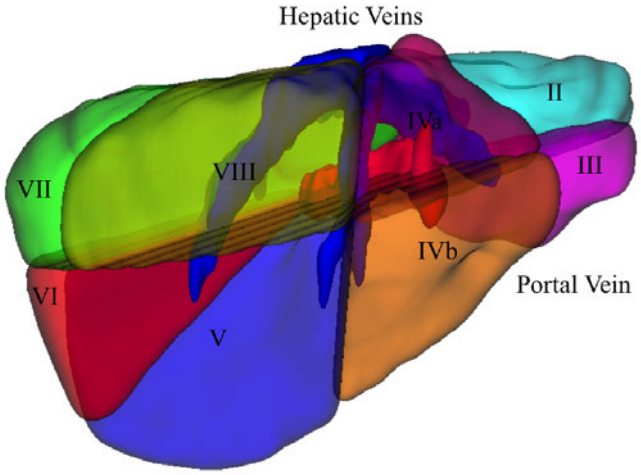


Figure 4.5.: Couinaud segments as calculated by fitting planes to the main branches of the portal and hepatic veins. Image taken from [OFC11].

was presented by Sorantin et al. [SWB*08].

In the system proposed by Debarba et al. [DZF*10], vessels are annotated by placing points on the vessels. Those points are then used to calculate the liver segments similar to the approach proposed by Selle et al. [SSPP00]. The actual planning step consists of selecting one or more segments to be removed and displaying volumetric information.

The method proposed by Oliveira et al. [OFC11] fits three approximately vertical planes to the main branches of the hepatic vein and one approximately horizontal plane to the orientation of the portal vein. The planes are fitted using voxels of the corresponding vessel branches in a least squares manner. An example of the result using this method is shown in Figure 4.5.

Drechsler and Oyarzun [DOL12a] presented a planning system that uses Selle et al.'s [SSPP00] approach to simulate the effect of intraoperative portal vein clamping. This technique is used by surgeons to detect the real segmental boundaries during interven-

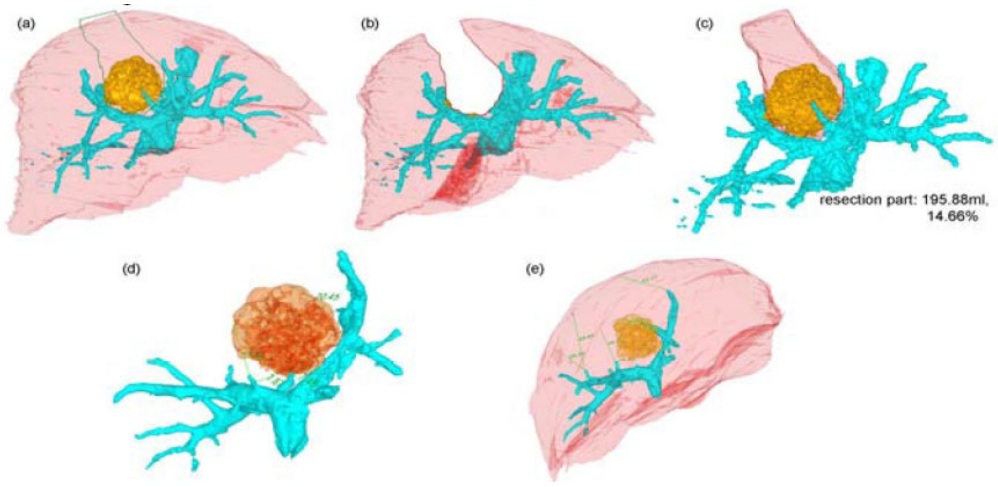


Figure 4.6.: Arbitrary resection lines drawn by the user to perform an atypical resection. Image taken from [SCW*11].

tion. By clamping the vein, the dependent parenchyma changes its color and the surgeon can use a coagulator to mark the boundaries on the liver surface. Chouillard et al. [CGC10] gave an overview of vascular clamping techniques.

4.2.2. Atypical non-anatomic resections

In cases when the tumor is in the periphery of the liver or when liver tissue must be preserved, the surgeon can perform a non-anatomic resection. Hereby, the tumor is resected with an arbitrary cut along a safety margin around the tumor. In comparison to anatomical resections, main vessels can cross the cutting line. Thus, a careful preparation is necessary to determine, if a non-anatomic resection is feasible. This mainly depends on possible safety margins without cutting major hepatic vessels. Each cut portal vein causes bleeding and an undersupply of parts of the liver, which can become necrotic. On the other side, each cut hepatic vein causes a loss of drain for some parts of the liver. It is also not unlikely that a tumor is bigger than assumed and often found during a surgery. Thus, it is advantageous to estimate the risk of a resection with respect to different safety margins around a tumor. The bigger the tumor and the bigger the safety margin, the more vessels are affected that supply or drain blood to/from parts of the liver.

Computer-based planning systems support the surgeon in different ways. Resections of peripheral tumors can be simulated using tools that allow to 'draw' resection lines, either with a specific shape or arbitrarily shaped. Other tools allow to assess surgical risk by means of visualizing affected vessels and/or parenchyma and provide quantitative measurements.

The system proposed by Glombitza et al. [GLD*99] allows to visualize vessels inside a security margin with a different color.

Numminen et al. [NSM05] presented a system that allows to use straight planes to divide the liver into two parts.

Konrad-Verse et al. [KVPL04] developed a deformable cutting plane for virtual resections. The basic idea of their method is to convert resection lines drawn on the surface

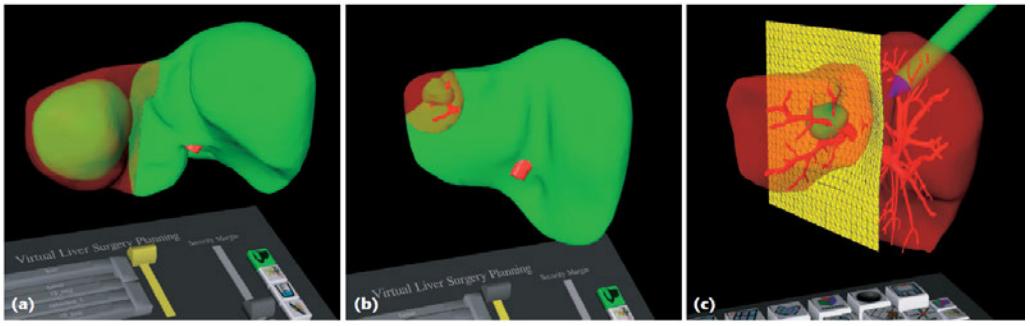


Figure 4.7.: Simulating atypical resections using different cutting tools. From left to right: A straight plane, a scalable sphere, and a deformable plane. Image taken from [RBBS06].

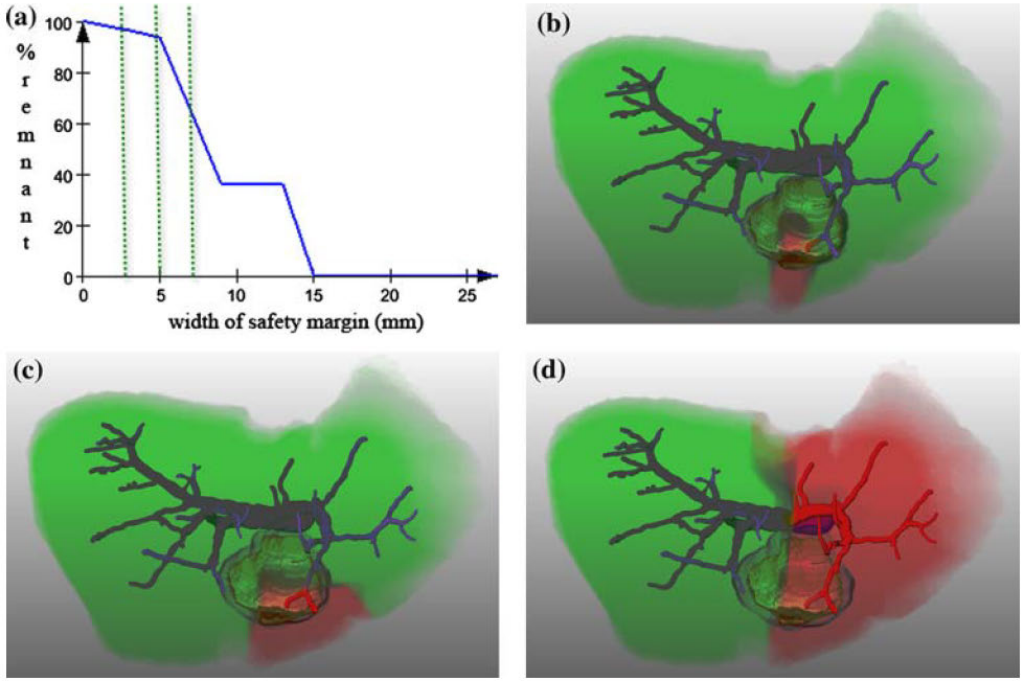


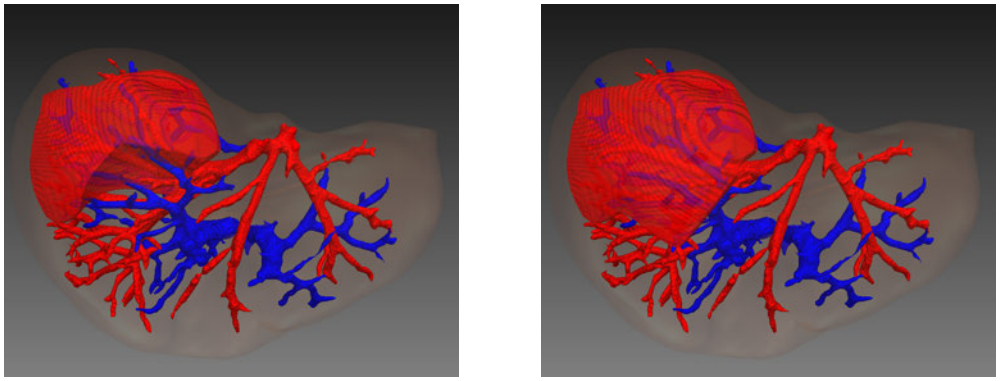
Figure 4.8.: Plotting remnant liver volume versus width of safety margin with dedicated 3D visualization. Image taken from [HZH*09].

of a liver model into an initial mesh, which can be further deformed with a sphere shaped tool.

The augmented reality planning system by Reitinger et al. [RBBS06] provides three tools for non-anatomical resection (Figure 4.7). A plane for straight cuts through the liver (multiple planes can be used for a resection), a scalable sphere for resections of tumors at peripheral locations and a deformable plane for complex cases. The authors noted that for most cases, it is sufficient to define straight paths using one or two planes.

Song et al. [SCW*11] developed a system where the surgeon can draw arbitrary resection lines to resect the tumor (Figure 4.6). Furthermore, they provide tools for quantitative analysis.

Hansen et al. [HZH*09] proposed a tool to determine robust safety margins around the



(a) Safety margin = 10mm (b) Safety margin = 15mm

Figure 4.9.: (a) Affected portal venous territory if the tumor is resected with a safety margin of 10mm. (b) Affected portal venous territory if the tumor is resected with a safety margin of 15mm. Image taken from [DEOLW12].

tumor. Therefore, their method analyzes cut vessels and their sub-branches using different safety margins. It then calculates corresponding portal and hepatic venous territories using Selle et al.'s [SSPP00] approach. The remnant liver volume versus safety margin is plotted in a graph and shows the optimal safety margin with respect to the liver remnant (Figure 4.8(a)). Affected vessels and territories are visualized in 3D (Figure 4.8(b-d)).

Shevchenko et al. [SSS*10] and Schwaiger et al. [SMS*10] presented a system that fully automatically segments liver vessels and performs risk analysis by means of visualization of affected vessels within three predefined margins around the tumor.

Drechsler et al. [DEOLW12] presented a planning system that performs a deformable registration of multiple CT acquisition phases in order to fuse complementary information. Concretely, tumors visible in the arterial phase can be fused with hepatic veins visible in the venous phase. Afterwards surgical risk can be assessed by detecting affected vessels within a safety margin around the tumor and visualization of the corresponding portal and hepatic venous territories. Figure 4.9(a) shows the affected portal venous territory if the tumor is resected with a safety margin of 10mm. Figure 4.9(b) shows the same situation, but with a safety margin of 15mm. It can be seen that a slightly larger safety margin affects a bigger area of the liver.

4.2.3. Discussion

Recent developments of computer-based systems for interventional planning of liver resections were reviewed in the previous sections. They can be classified according to their medical use case into systems for anatomical or non-anatomical resection planning. The former allows for the calculation of functional liver segments, which can be calculated based on annotated hepatic vessels or based on planes fitted to the main hepatic branches. The latter provides tools to simulate arbitrary resections and perform risk analysis based on individual safety margins around the tumor. Proposed systems mainly differ in the used algorithms, the degree of automatism and the provided planning tools integrated into the application. Some systems do not offer methods to separate

	Data preparation					Planning tools			
						Typical		Atypical	
Publication	Organ	Vessels	Separation	Tumor	Fusion	Vessels	Planes	Arbitrary	Risk analysis
Glombitza et al. [GLD*99]	Basic tools	Threshold	Graph	Basic tools	n/a	•	•		•
Selle et al. [SSPP00]	Livewire	Region	Graph	n/a	n/a	•			
Soler et al. [SDM*01]	Model	Threshold	Graph	Threshold	n/a	•			
Meinzer et al. [MSS*04]	Basic tools	Region	Graph	Basic tools	Affine	•			
Reitinger et al. [RBBS06]	Model	Region	n/a	Model	n/a	•		•	
Debarba et al. [DZF*10]	Livewire	n/a	n/a	n/a	n/a	•			
Oliveira et al. [OFC11]	Model	Region	Voxel	Region	n/a		•		
Drechsler & Oyarzun [DOL12a]	Model	Region	Graph	Sphere	n/a	•			•
Numminen et al. [NSM05]	Threshold	Manually	n/a	Manually	n/a			•	
Konrad-Verse et al. [KVPL04]	n/s	n/s	n/s	n/s	n/a			•	
Song et al. [SCW*11]	Level set	Region	n/a	Region	n/a			•	
Hansen et al. [HZH*09]	Livewire	Region	Graph	Region	n/a				•
Shevchenko et al. [SSS*10]	Region	Region	n/a	Region	n/a				•
Drechsler et al. [DEOLW12]	Model	Region	Graph	Graph-cut or Radial rays	Deformable				•

Table 4.1.: Summary of discussed publications. Algorithms for data preparation and provided planning tools are listed. n/a: not available by the described system. n/s: supported, but not specified in the paper. Table taken from [DOL12b].

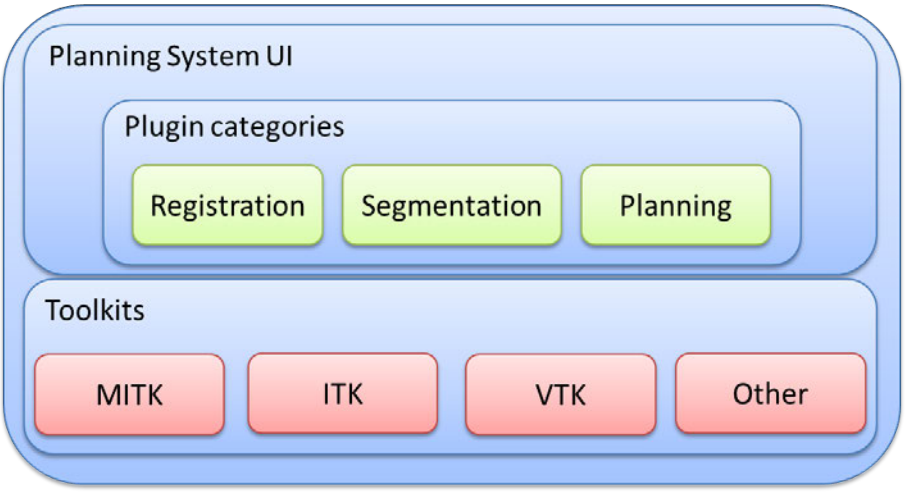


Figure 4.10.: Architecture of the proposed system. Image taken from [DEOLW12].

interconnected vessels, others do not provide the fusion of information spread in multiple datasets. Table 4.1 provides a summary of the publications mentioned in this work. It contains information about used methods for data preparation and which tools are provided for resection planning. The former is divided into organ, vessel and tumor segmentation methods. Usually, these structures are segmented using variations of threshold-, region- or model-based approaches. But also manual methods (drawing in each slice) were used. Meinzer et al. [MSS*04], for example, provides a set of basic tools, like region growing and active contours. Furthermore, the table lists if vessels are separated using a graph representation or if it is solely based on voxel information. If data fusion is provided, it is listed if the registration process is performed in a rigid-, affine or deformable manner.

4.2.4. Conclusion

Although interventional planning systems reached a stable state and are already used in clinical practice, the full potential of computer assisted liver resection planning is not unleashed yet. Several researchers are still actively investigating ways to improve and extend possibilities. Current trends include the fusion of information from multiple datasets and the automation of various steps in the workflow. Hansen et al. [HLZ*10], for example, investigated ways to automatically generate resection proposals. This is quite challenging due to the lack of expert knowledge. For instance, surrounding structures and deformation must be taken into account to plan access to the tumor. Furthermore, a fully automatic segmentation of all relevant structures is highly desired to save time and reduce inter- and intraobserver variability.

4.3. Overview

For a planning system, the whole processing chain is crucial and not only single modules. The application developed in this work consists of several independent plugins and implements a data-driven workflow, e.g. if module 'B' depends on data produced by module

'A', then it will refuse user input, if data produced by module 'A' is not available. Each plugin is autonomous and decides which data is visualized as soon as it is activated. These simple rules kept dependencies and communication overhead low and allowed for an independent development of plugins by different researchers. Implementation effort was further reduced by using several open source toolkits, like MITK[1], ITK[2], VTK[3] and QT[4], but also commercial toolkits like Intels MKL[5]. Figure 4.10 illustrates the architecture of the proposed system.

Concretely, the application consists currently of six plugins. Each one for organ segmentation, deformable registration, tumor segmentation, vessel extraction & abstraction, Couinaud segmentation and risk analysis. As illustrated in Figure 4.10, each plugin can be assigned to one plugin category: Registration, segmentation, or planning .

The organ segmentation plugin allows for an automatic segmentation of the liver using an automatic approach based on statistical shape models [EKSW10]. Furthermore, it provides 3D tools for manual refinement, if necessary.

The deformable registration method used in this work to register portal venous CT scans of the liver with arterial phase scans is based on the work of Erdt et al. [ESH*11]. In their approach, the liver is first segmented in both arterial and portal venous phases. The liver shapes are then registered by a landmark based registration to bring both images in a rough alignment. Afterwards, a deformable voxel based registration scheme is applied. This scheme takes the distance of a voxel to the liver boundaries into account in order to avoid a false matching of non-corresponding structures inside of the liver. This is important, since in both phases different vessel structures are enhanced that should not be matched. Figure 4.11 shows an exemplary result of the deformable registration.

A drawback of the original approach was the runtime of the registration part. A single run takes about 22 minutes on a 2.93 GHz Intel Quad Core processor. Its implementation was modified in order to speed up the execution time by the same author in [DEOLW12]. Therefore, an image pyramid with 3 resolution levels is applied. The coarsest level has an isotropic resolution of 4.0 mm voxel spacing, the voxel spacing of the second level is 2.0 mm and the last level has the original voxel spacing of the image. 20 iterations of the coarsest level and 10 iterations for the remaining levels turned out to give the best speedup while preserving the registration quality. As a second modification, the speed function is only evaluated for voxels that have a distance below 30 mm to the boundary. In multi-processor environments, this usually reduces the workload of each core by 20 to 30%, because the time for processing single regions vary. However, since less voxels are processed, a significant speedup can be achieved.

By using the described modifications, the runtime of the whole method could be reduced from 22 minutes to 6 minutes on the same machine. Here, around 3 minutes of the total processing time is taken by the segmentation part which has not been accelerated in this work. The runtime of the registration part could therefore be enhanced by a factor of 6.

[1] http://www.mitk.org
[2] http://www.itk.org
[3] http://www.vtk.org
[4] http://qt.nokia.com
[5] http://software.intel.com/en-us/articles/intel-mkl

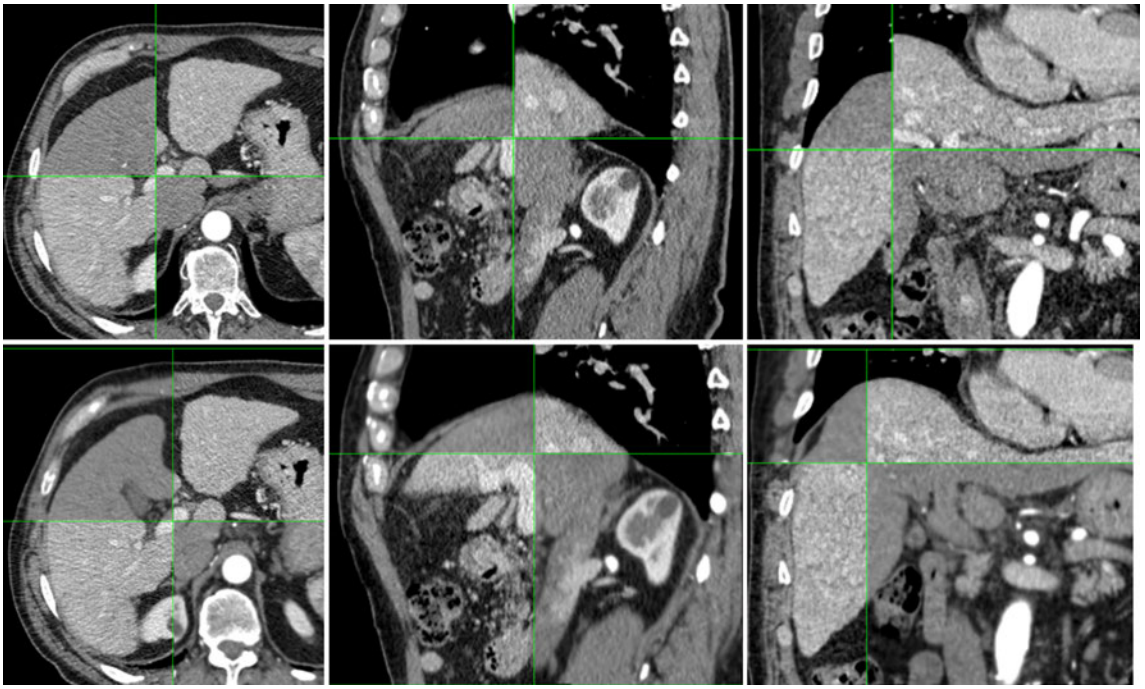


Figure 4.11.: Registration of arterial CT contrast phase with portal venous phase of the liver. The top row shows the alignment before registration and the bottom row shows the alignment after deformable registration. Image taken from [DEOLW12].

The tumor segmentation plugin provides two tools to segment tumors. The first one is an interactive 3D radial ray based approach, which uses belief propagation to optimize the contours [SS12]. The second one is a graph-cut based approach [DSOL11] and is described in detail in Section 4.4.

The vessel extraction & abstraction plugin allows for a semi-automatic extraction of hepatic veins using the methods described in Chapter 3. Namely, segmentation of vessels using a Hessian-based multi-scale analysis approach [DOL10d], creation of formal graph structures [DOL10c] and separation of portal and hepatic veins by performing a graph-based analysis [DOOL11].

The Couinaud segmentation plugin provides tools to annotate branches of the portal vein according to their segment membership and approximation of liver segments using a Voronoi tesselation [SPSP02].

The risk assessment plugin allows to estimate the surgical risk if the tumor is resected with a specific safety margin. Hereby, it is crucial that blood supply and drain is not suppressed for parts of the remnant liver, which would result in necrotic liver tissues. Furthermore, a specific amount of remnant liver should remain to ensure a positive outcome for the patient. The analysis of different scenarios based on safety margins is part of this module. This is done by analyzing the vessel structure around a tumor. It will be explained in detail in Section 4.5.2.

4.4. Tumor segmentation based on graph cuts

Modern computer assisted liver surgery planning requires the segmentation of all relevant structures like the organ, vessels and tumors. The key challenge for the latter is to determine the exact location and volume. Difficulties arise because of low intensity boundaries, varying shapes and sizes. Furthermore, they can be located everywhere in the liver. Interactive segmentation methods seem to be the most appropriate in terms of reliability and robustness in this case. Indeed, an interactive tumor segmentation method based on graph-cuts was the winner of the MICCAI challenge 2008 [SDB08]. Graph-cut based methods usually find globally optimal segmentations. However, the complexity of the underlying graphs is enormous for clinical 3D datasets. For example, when using the library provided by Kolmogorov[6] the memory consumption of a node is 48 byte and 32 byte for an edge on a 64-bit system. Thus, a CT image with 512x512x134 voxels requires around 8 GB to be represented as a graph using a 6-Neighborhood and around 30 GB using a 26-Neighborhood.

Graphs can be reduced by either manually cropping a region of interest or by applying automated methods. Stawiaski et al. [SDB08] used watershed to group homogenous regions, each represented by a single node in the graph. However, in their method they also rely on a manually selected region of interest that includes the tumors to be segmented. Furthermore, experiments showed that applying watershed is slow and introduces additional parameters that must be fine tuned to get optimal results.

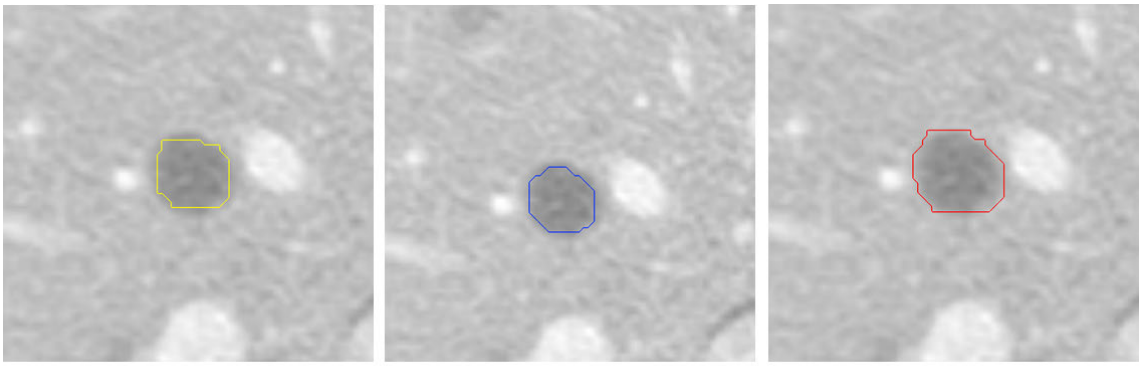
Lombaert et al. [LSGX05] reduced graph complexity by using an image pyramid and a narrow band along the boundaries from the segmentation result of one level as initialization for the next level. The authors have shown that the result is comparable with a conventional graph-cut method. However, using only a narrow band on the next level assumes that everything inside the narrow band is already correctly segmented, which could cause a loss of detail. Furthermore, in cases where the boundary of the object to be segmented is "spiky", the results deviate from the results produced by a conventional graph-cut method.

In the following sections, a method is presented to identify a region of interest (ROI) in the original image to reduce graph complexity for liver tumor segmentation using graph-cuts. The proposed approach uses a coarse resolution image to determine a ROI that is used as initialization to construct a reduced graph for final segmentation in the original image. Evaluation results suggest that the accuracy is comparable to other approaches with an improved computational and memory efficiency.

4.4.1. Graph-Cuts segmentation

Let $I^N = (P, f)$ be a discrete image of dimension N consisting of a finite set of $N - D$ points P and a function $f : P -> Y$ that maps a point $p \in P$ to an intensity value in an appropriate number space Y. For such an image a graph $G = (V, E, c)$ consisting of a set of nodes $V = P \cup \{s, t\}$, a set of edges $E = E_n \cup E_t$ and a cost function $c : V \times V \rightarrow R^+$ that defines the weight of the edges is constructed. The special nodes s and t define the object

[6] http://www.cs.ucl.ac.uk/staff/V.Kolmogorov/software.html



a b c

Figure 4.12.: The effect of individual terms in Equation 4.4. (a) Result using Equation 4.4 without the gradient term. (b) Result using Equation 4.4 without the intensity term. (c) Result using Equation 4.4. Image taken from [DSOL11].

terminal node and the background terminal node respectively. E_n are neighborhood links (n-links) and E_t terminal links (t-links).

The cost for a segmentation A can be calculated as

$$E(A) = \lambda \cdot R(A) + B(A) \tag{4.3}$$

where $R(A)$ is a regional term that takes the probability of a voxel belonging to the object into account and λ a weight factor for this term. $B(A)$ is a term that takes the boundaries of an object into account. The problem is now to find a segmentation A_o such that $E(A_o)$ is minimal. It is known that $E(A)$ can be minimized by calculating a min-cut in a graph. In this work, the max-flow/min-cut algorithm of Boykov and Kolmogorov [BK04] is used. Because the goal is a generic segmentation tool where the probability distribution of object voxels is not known in advance or is ambiguous, the regional term is disabled by setting $\lambda = 0$. The boundary term is defined as

$$B(A) = \sum_{\{p,q\}\ inN} (exp(-\frac{(f(p) - f(q))^2}{2\sigma^2}) \cdot \frac{1}{\|p_{phy} - q_{phy}\|} \cdot exp(-(g(p) - g(q)))) \tag{4.4}$$

where $g : V \rightarrow R$ is the gradient magnitude and N a set with all unordered pairs of neighbors. Figure 4.12 visualizes the effects of the intensity and gradient terms on the segmentation result. The distance term $\frac{1}{\|p_{phy} - q_{phy}\|}$ is used to penalize neighbors that are far away from each other, as it is the case for example in anisotropic datasets. In Figure 4.13 it can be seen that the distance term has a positive effect on the segmentation result.

4.4.2. Automatic identification of a region of interest

The basic idea to determine automatically a region of interest (ROI) is to apply the graph-cut algorithm on a lower resolution version of the original image. The result is then used to constrain the graph creation for the original image. The advantage is that graph complexity is greatly reduced, while details are preserved in the final segmentation. The proposed

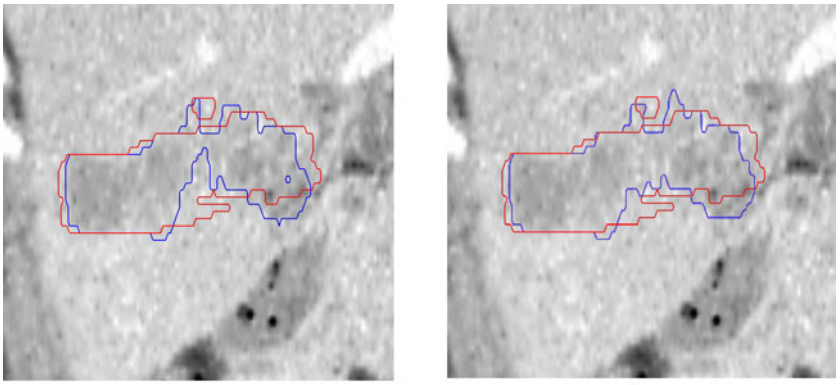


Figure 4.13.: Effects of the distance term. Ground-truth segmentation (red) vs. result of the graph-cut segmentation (blue) without (a) and with (b) distance term. Image taken from [DSOL11].

method is executed in the following steps.

Step 0 The user has to provide two discrete images $O^N = (P, f)$ and $B^N = (P, f)$ where $f : P \rightarrow \{0, 1\}$ marks a point $p \in P$ with $f(p) = 1$ as object and background seed respectively. In the following, the set of object seeds is called o_n and the set of background seeds b_n.

Step 1 Resize the image by combining $n \times n \times n$ voxels and represent this combination by the mean intensity. Furthermore, resize the images with user selected object and background seeds in the same way. For evaluation, this parameter was set to $n = 3$.

Step 2 Create a directed graph for this lower resolution image. E_n is built by creating n-links (p,q) between neighboring voxels $p, q \in V$ with a weight/capacity calculated by the cost function $c(p, q)$ which is derived from Equation 4.4 as

$$cost(p, q) = exp(-\frac{(f(p) - f(q))^2}{2\sigma^2}) \cdot \frac{1}{\|p_{phy} - q_{phy}\|} \cdot exp(-(g(p) - g(q))). \tag{4.5}$$

E_t is built by creating t-links (s, o_n) and (b_n, t) between terminal node s and object seeds o_n and t and background seeds b_n. The capacity of t-links is set to infinite. In this step, the regular grid structure of an image was utilized to uniquely and linearly map voxels to nodes (Figure 4.15 (a)). This mapping can be calculated as

$$Node_{ID} = x_{index} + (y_{index} \cdot x_{size}) + (z_{index} \cdot x_{size} \cdot y_{size}), \tag{4.6}$$

where x_{index}, y_{index} and z_{index} denote the current position and x_{size}, y_{size} and z_{size} the maximal extend of the volume in each direction.

Step 3 Apply the graph-cut algorithm to this graph to obtain a rough segmentation.

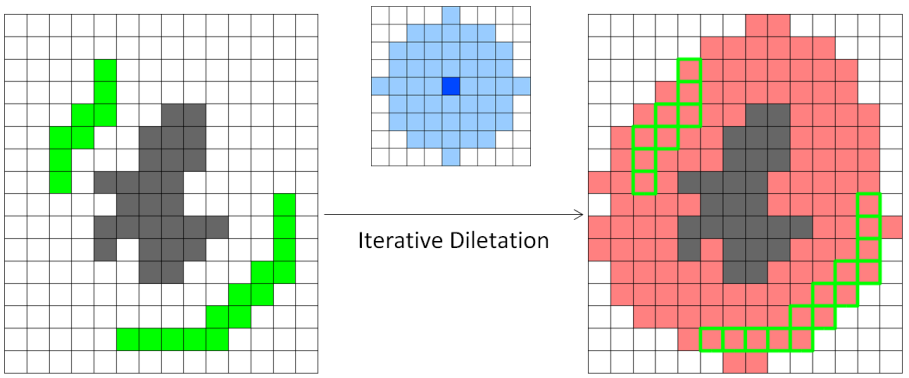


Figure 4.14.: Iterative dilation of the rough segmentation result (grey) with a ball structuring element (blue) until all background seeds (green) are included in the result. Image taken from [DSOL11].

Step 4 Postprocess the result from the previous step by blowing it up using morphological dilatation as illustrated in Figure 4.14. Basically, the goal is to achieve that the rough segmentation is bigger than the object to be segmented and that there are no holes and cavities within the segmentation to preserve details. Furthermore, the user selected background seeds must be included. The result of this step is used as the ROI in the original image. Experiments showed that a structuring element with 4 voxels diameter was sufficient for the dilatation operator (for n=3 in Step 1). The result of this step is used as the ROI in the original image.

Step 5 Create a graph that covers the volume in the original image defined by the ROI from the previous step. In this step, voxels cannot be mapped to node IDs using the Equation 4.6 from Step 2 anymore, because the ROI is not regularly arranged. Therefore, each voxel in the ROI is consecutively numbered (Figure 4.15 (b)). This structure is used to determine the IDs for the nodes in the graph and to find out which voxels are in the neighborhood to create n-links. Weights/capacities are calculated as in Step 2. Again, t-links are created as in Step 2. However, this time using the original seeds and not the resized ones. Using the ROI from the previous step greatly reduces the size of the graph without losing details in the final segmentation.

Step 6 Apply the graph-cut algorithm again to obtain the final result.

In case of unsatisfying results, the user can add or remove seed points and repeat the process to iteratively correct the results as shown in Figure 4.16.

4.5. Planning of typical and atypical resections

4.5.1. Vessel-based and plane-based Couinaud segmentation

To simulate the color change on the surface of the liver after clamping the portal vein at a specific position, the Nearest Neighbor Approach originally proposed by Selle et

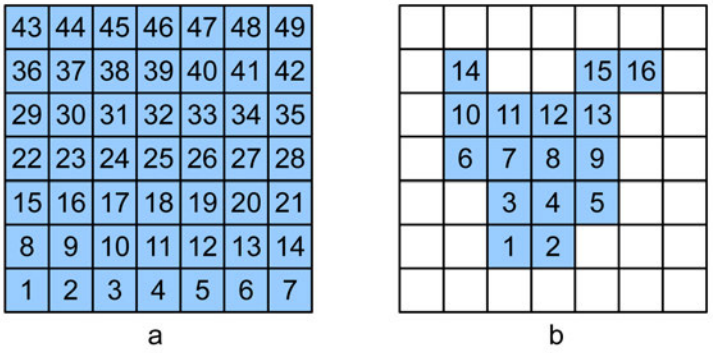


Figure 4.15.: (a) To define a mapping between nodes in the graph and voxels in the image the regular grid is used. (b) Because the ROI is not regulary arranged, each voxel is consecutively numbered and used for graph creation. Image taken from [DSOL11].

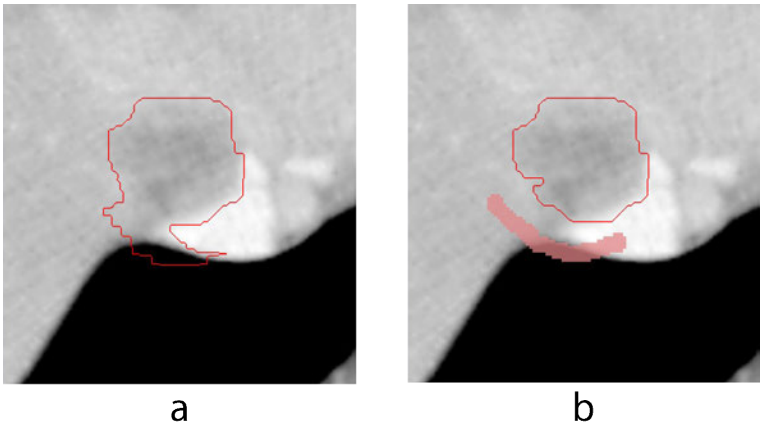


Figure 4.16.: Interactive correction of unsatisfying results by adding additional seed points. (a) The segmentation shows a leakage. (b) The result was corrected by adding additional background seeds. Image taken from [DSOL11].

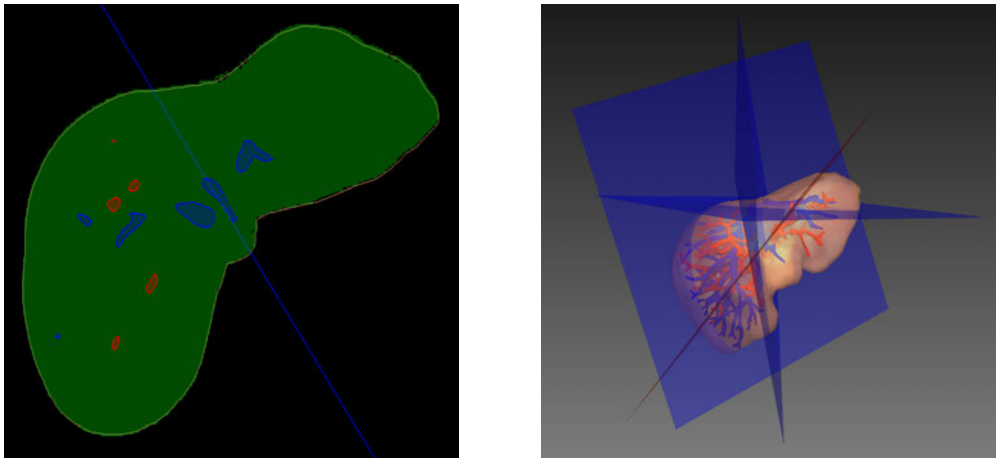
al. [SPSP02] is used to classify each liver tissue voxel. Let L be the set of liver voxels and P_j a portal vein branch with label j, $j = 0, .., n-1$. Function $g(x, y)$ labels voxel x with label y and $dist(x, y)$ calculates the Euclidean distance between voxel x and y. Then

$$\forall v \in L.g(v, f(v)), \tag{4.7}$$

with

$$f(v) = \underset{j=0,\ldots,n-1}{\operatorname{argmin}} \; \underset{v_i \in B_j}{\operatorname{argmin}} \, dist(v, v_i). \tag{4.8}$$

The result is a patient-specific approximation of the portal venous territory that is supplied starting at the clamping position. During an intervention, the surgeon can clamp the portal vein at a position that supplies blood to the (sub-)segment where the tumor is located. This leads to a color change on the surface of the liver, which indicates the borders of this (sub-)segment. The surgeon then usually uses a coagulator to 'draw' the borders on the



(a) (b)

Figure 4.17.: Couinaud liver segmentation. (a) One plane through the middle branch of the liver vein. (b) Three planes along the main liver vein branches and one plain along the portal vein.

liver surface before releasing the clamped vein. Selle et al. [SPSP02] showed that patient-individual liver segments can be approximated by using the portal vein and mathematical methods. They evaluated the accuracy of this method to be between 80-90% overlap with respect to the real segments.

Patient-specific Couinaud segments are yet to be established in clinical practice. Traditionally, three horizontal and one vertical planes are drawn along the major branches of the liver and portal vein systems. A possibility to place rigid planes in order to calculate the Couinaud segmentation was also implemented. Basically, for each plane the user has to add seed points in the dataset to define the plane. The plane is then visualized in the 2D views (Figure 4.17(a)) and additionally in 3D (Figure 4.17(b)). If the user is satisfied, the planes are used to cut the liver into pieces.

4.5.2. Risk assessment of atypical resections

As reviewed in Section 4.2, systems for image-guided planning of liver resections were published.

Besides individual shortcomings in one or more of the mentioned modules, these systems do not take multiple phases of the acquired CT volume into account. However, there are tumors only visible in the arterial phase, while other tumors and the important hepatic veins are only visible in the venous phase. The proposed system uses fused information from both, arterial and venous phases. Furthermore, a high degree of automation is reached for all relevant steps.

4.5.2.1. Risk assessment

To judge the feasibility of a planned resection, a visual component was implemented that draws an arbitrarily shaped surface, called safety margin, equally spaced around the tumor with a user-specified distance. Usually, a safety margin of 1 cm is said to be safe.

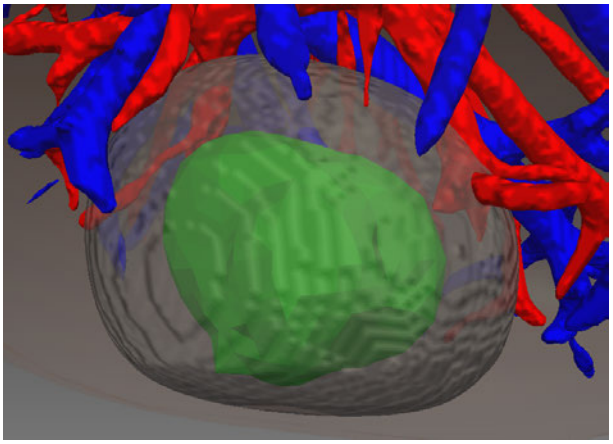


Figure 4.18.: The safety margin is rendered as surface for simplifying visual inspection of affected vessels.

However, the tumor can be bigger compared to what is visible in a CT dataset. Thus, by adjusting the safety margin, different scenarios can be evaluated. Together with a rendering of the hepatic veins, it can be visually perceived which vessels are affected if the provided safety margin is strictly followed during a resection.

As each segmented tumor is stored in a volume with the same dimension as the original dataset, an automatical cropping is performed in order to reduce the data and to speedup the following calculations. The bounding box around the tumor is chosen in such a way that the specified safety margin fits inside. Next, a distance map of the cropped tumor volume is calculated and thresholded to create the requested safety margin. Then, all voxels are deleted that are outside the liver. Finally, a 3D visualization is obtained by applying the Marching Cubes algorithm. An example is shown in Figure 4.18.

To further help with the assessment of risk, the supply area of the portal vein and drain area of the hepatic vein are automatically calculated, depending on the vessels affected by the user-specified safety margin. A sketch of the algorithm is provided by Algorithm 1. Several data structures are utilized which are explained as follows.

- veinSkeleton: This is a volume containing the skeleton of the portal and hepatic veins. Its voxels are annotated as either affected or not affected and used for Voronoi tesselation together with a mask of the segmented liver (liverMask).

- labeledSkeleton: This is a volume containing a labeled skeleton of either the portal or hepatic veins. Each voxel is labeled with an ID which corresponds to an edge in a graph representation of the same vessel tree. It acts as a link between voxel and graph representation of a vessel tree.

- labeledVein: This is a volume containing labeled segmentations of either portal or hepatic veins. It is created by propagating the labels of the labeledSkeleton within the lumen of the segmented vessels. Thus, links between skeleton, segmentation and graph representations are established.

A highly optimized k-d tree data structure to perform range searches in order to find affected vessels is used [ML09]. Furthermore, a k-d tree is used as the main data structure

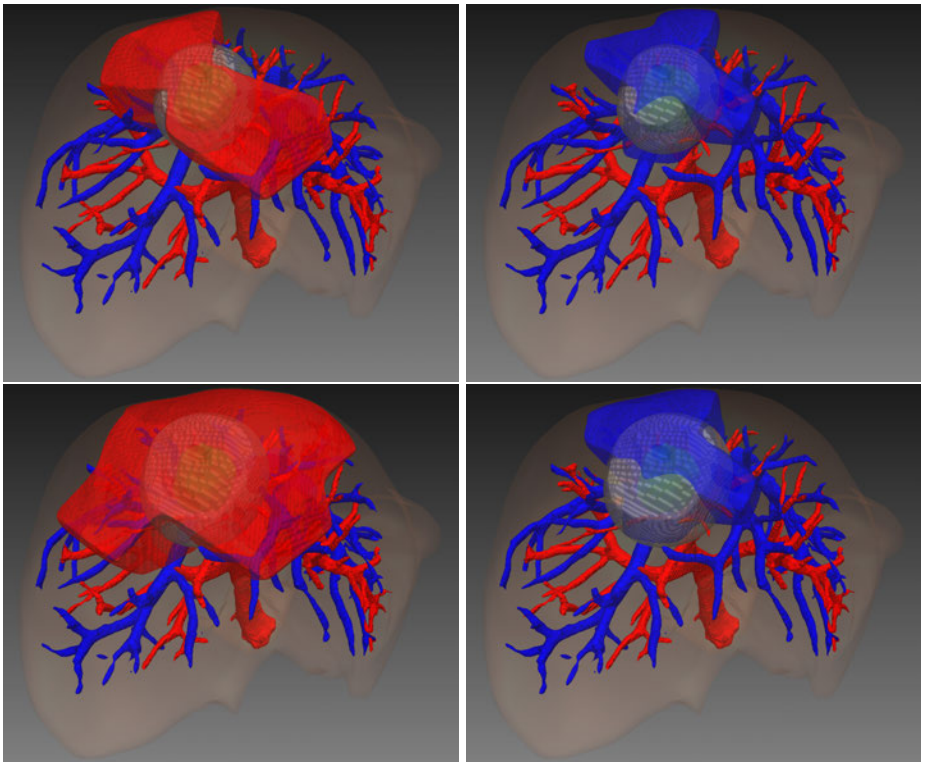


Figure 4.19.: Top row: Affected supply (red) and drain (blue) territories with a 1.0 cm safety margin. Bottom row: Affected territories with a 1.5 cm safety margin.

to implement the nearest neighbor search for Voronoi tessellation.

Algorithm 1 Calculation of supply and drain areas for each vein

```
procedure CalculateTerritory(...)
    kdTree.init( labeledVeinBorderPoints )
    for all tumorSurfacePoints n do
        affectedbranchIDs ← affectedbranchIDs + kdTree.rangeSearch( n, dist )
    end for
    for all affectedBranchIDs n do
        subTreeIDs ← GetSubTreeIDsFromGraph( n )
    end for
    IDs ← affectedBranchIDs + subtreeIDs
    Annotate( IDs, veinSkeleton )
    VoronoiTesselation( liverMask, annotatedVeinSkeleton )
end procedure
```

Figure 4.19 shows an example of the affected portal and hepatic venous territories. For the top row, a safety margin of 1.0 cm was chosen. For the bottom row the safety margin was 1.5 cm. It can be seen that significantly more of the remnant liver would not be supplied with blood anymore, if the safety margin is increased by only 0.5 cm.

4.6. Results

In this section, results of the developed application are presented. Section 4.6.1 shows results of the proposed tumor segmentation method. Section 4.6.2 presents the planning workflow as implemented by the application in a step by step manner. Section 4.6.3 shows results of the risk assessment module and retrospectively analyzes two clinical cases.

4.6.1. Tumor segmentation

For evaluation, seven tumors in two contrast enhanced CT datasets of the liver were segmented with in-plane spacings between 0.52 and 0.78 mm and out of plane spacing between 1.6 and 4 mm. Ground-truth segmentations were done by experts. The tumors and ground-truth mask for a 2D slice are shown in Figure 4.20. All experiments were executed on a Windows 7 PC equipped with a 2.4 GHz Intel Core 2 Quad CPU and 4 GB RAM. The presented approach was implemented as a filter pipeline using the ITK framework[7].

The segmentation accuracy is evaluated using the Dice similarity coefficient defined as

$$Dice(P, Q) = (2 \cdot |P \cap Q|)/(|P| + |Q|) \tag{4.9}$$

where $Dice(P, Q) = 0$ means no overlap and $Dice(P, Q) = 1$ perfect overlap. Furthermore, the mean surface distance is used, which is defined as

$$Dist_{mean}(S_P, T_Q) = \frac{1}{n} \sum_{i=1}^{n} \min_{j=1,\ldots,m} \|S_P(j) - T_Q(i)\| \tag{4.10}$$

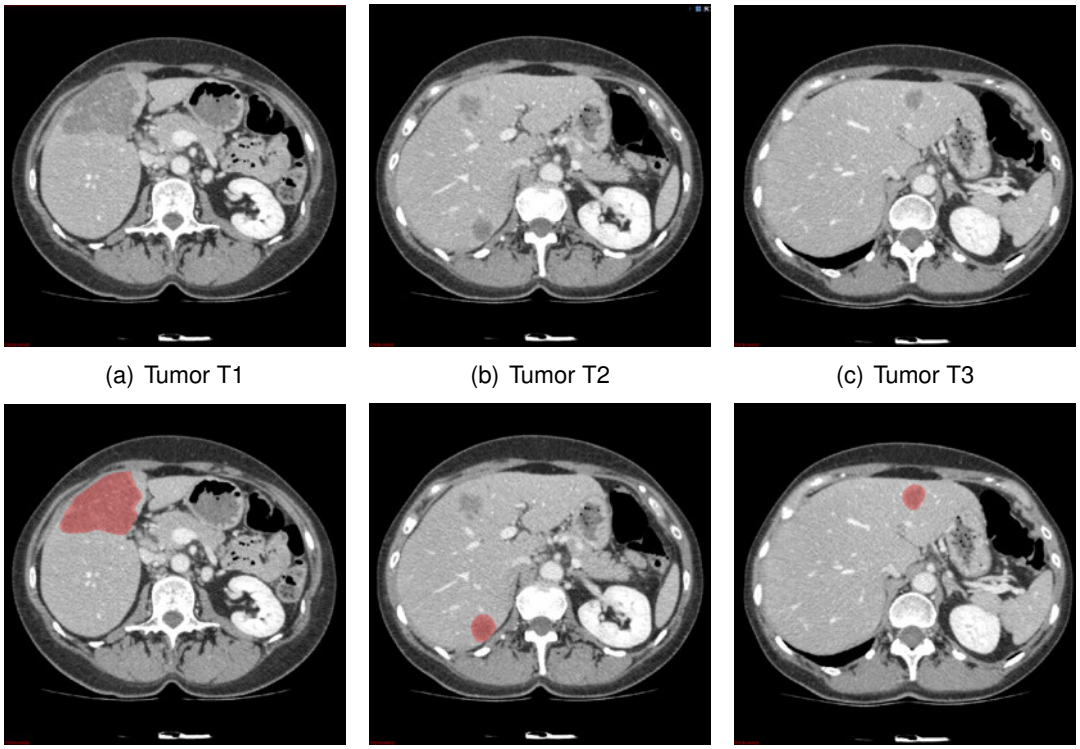
and the Hausdorff distance to calculate the maximum distance defined as

$$Dist_{max}(S_P, T_Q) = \max\{d_{max}(S_P, T_Q), d_{max}(T_Q, S_P)\} \tag{4.11}$$

with $d_{max} = max_{i \in S_P} \min_{j \in T_Q} \|i - j\|$. P is the ground-truth segmentation, Q the segmentation result and S_P and T_Q sets of surface point vectors.

Quantitative results using a 6-Neighborhood for graph creation are provided in Table 4.2 and visual examples are shown in Figure 4.21. A 26-Neighborhood was also used which had no significant accuracy improvements as can be seen in Table 4.3, but processing took 10 seconds more time and consumed around 4 times more memory. Stawiaski et al. [SDB08] got an average $Dist_{mean}$ of 1.5 mm, an average $Dist_{max}$ of 8.29 mm and an average overlap error of 29.49%. In comparison to the presented results (average overlap error: 20%, average surface distance: 0.73 mm, average maximum surface distance 5.31 mm), this is a little worse. However, it was not possible to get the same datasets that they used for their evaluation for the MICCAI challenge. Thus, a side-by-side comparison is not possible. However, it can be concluded that the proposed method produces similar, if not better, results.

[7] http://www.itk.org



(a) Tumor T1 (b) Tumor T2 (c) Tumor T3

(d) Mask of T1 (e) Mask of T2 (f) Mask of T3

(g) Tumor T4 (h) Tumors T5, T6, T7

(i) Mask of T4 (j) Masks of T5, T6, T7 (right to left)

Figure 4.20.: Tumors and ground-truth masks used for evaluation.

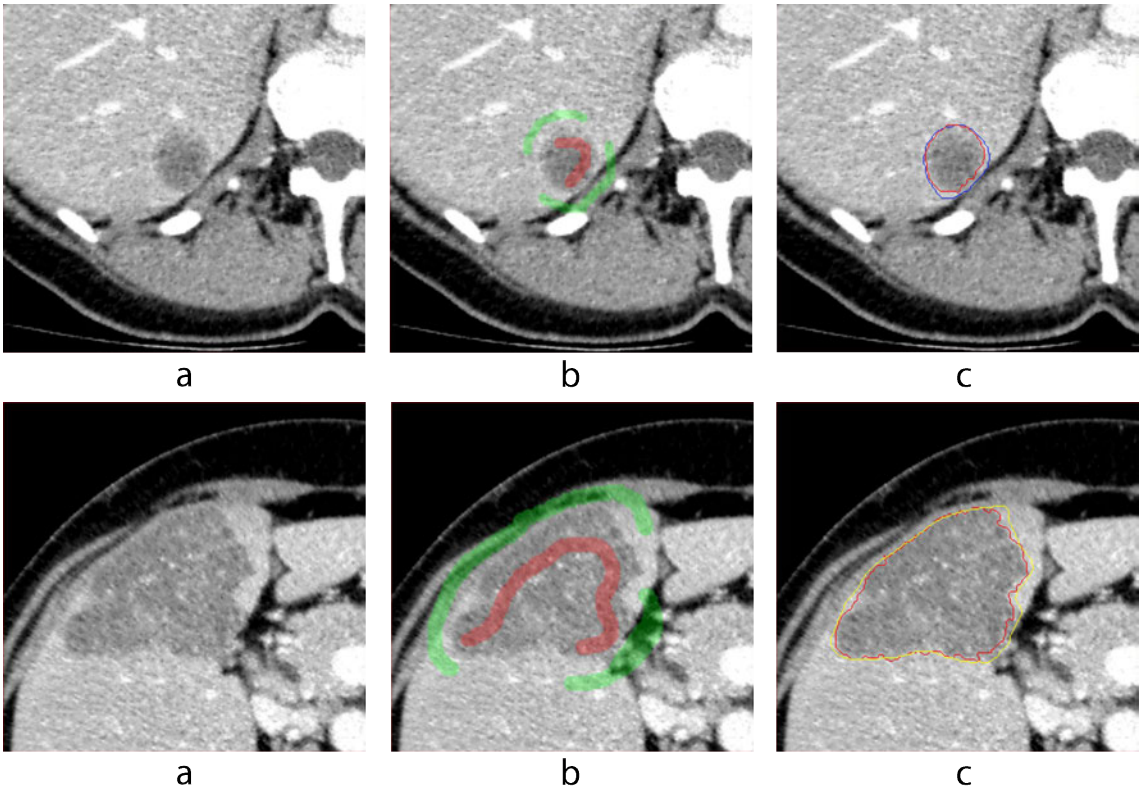
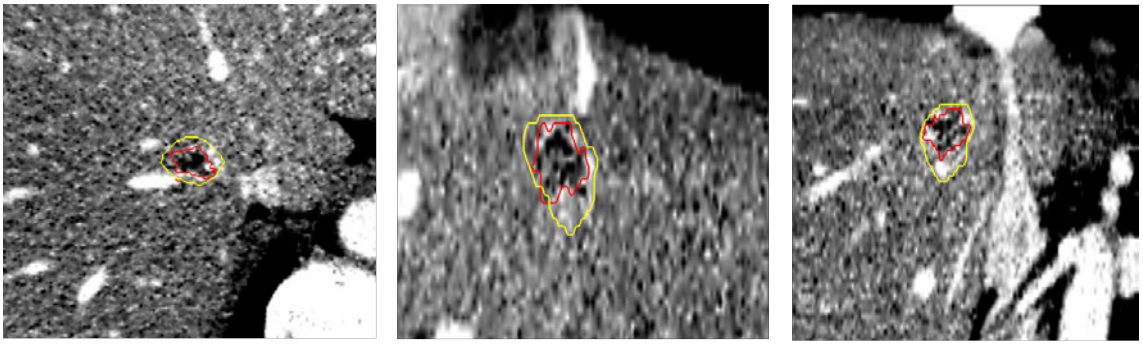


Figure 4.21.: (a) Tumor T2 (first row) and T1 (second row). (b) Object (red) and background seeds (green) selected by the user. (c) Segmentation result (red) and the ground-truth segmentation (yellow/blue). Image taken from [DSOL11].

Memory usage was greatly reduced using the described method and depends on the size of the object to be segmented and how much is segmented to identify a ROI. Leakings into neighboring areas quickly cause an increase in memory consumption. In one case, the tumor boundaries were hardly visible, which lead to a memory consumption of around 1 GB. However, during the experiments, memory consumption was in general very low as can be seen in Table 4.2 and 4.3. Figure 4.22 shows axial, sagittal and coronal view of the segmentation result of tumor T4, which is the worst result in the experiment with respect to the Dice metric (Equation 4.9). It can be seen that the dark area is quite well segmented. However, in this case experience of the expert is responsible that the ground-truth data covers a much bigger volume, which also includes bright areas.

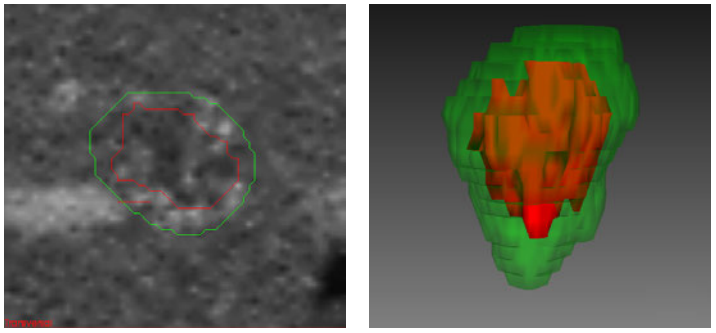
Figure 4.23 shows ground-truth and segmentation results for tumor T4, which produced the worst result. It can be seen that the ground-truth segmentation contains several dark and bright patches. The tumor is not clearly delineated, but edges are needed to produce a satisfying result.

The average overall segmentation time was 97 sec. Around 70 secs. of the time were spent for image preprocessing (noise reduction, gradient magnitude). Around 27 secs. were spent for automatic ROI detection and the following segmentation in the original image.



a b c

Figure 4.22.: The worst segmentation result (red) compared to ground-truth data (yellow) was produced for tumor T4. (a) Axial view. (b) Sagittal view. (c) Coronal view. Image taken from [DSOL11].



(a) (b)

Figure 4.23.: Worst case scenario (Tumor T4). (a) It can be seen that the tumor is not clearly delineated. The ground-truth segmentation (green) contains several dark and bright areas. (b) The segmentation result (red) is much smaller than the ground-truth.

4.6.2. Planning - workflow

In this section, the results of the developed tool at each stage of the application workflow from a user point of view are described. The procedure consists of two stages. In the first stage, the necessary preprocessing of the data is carried out. The second stage deals

Tumor	*Dice*	$Dist_{mean}$ [mm]	$Dist_{max}$ [mm]	Memory [MB] (V / E)
T1	0.916	0.961	5.209	26 / 103
T2	0.879	0.606	3.317	3 / 15
T3	0.879	0.442	2.350	3 / 12
T4	0.567	0.567	7.278	2 / 9
T5	0.784	0.826	4	1 / 5
T6	0.747	1.226	8	206 / 824
T7	0.835	0.499	7.044	10 / 42
Average	0.801	0.732	5.314	

Table 4.2.: Quantitative results using a 6-Neighborhood. Table taken from [DSOL11].

Tumor	*Dice*	$Dist_{mean}$ [mm]	$Dist_{max}$ [mm]	Memory [MB] (V / E)
T1	0.911	1.012	5.157	26 / 463
T2	0.878	0.601	3.461	3 / 69
T3	0.872	0.432	2.350	3 / 52
T4	0.563	1.638	8.160	2 / 27
T5	0.799	0.815	4	1 / 24
T6	0.734	1.103	8	206 / 3573
T7	0.913	0.476	6.367	10 / 184
Average	0.811	0.87	5.357	

Table 4.3.: Quantitative results using a 26-Neighborhood. Table taken from [DSOL11].

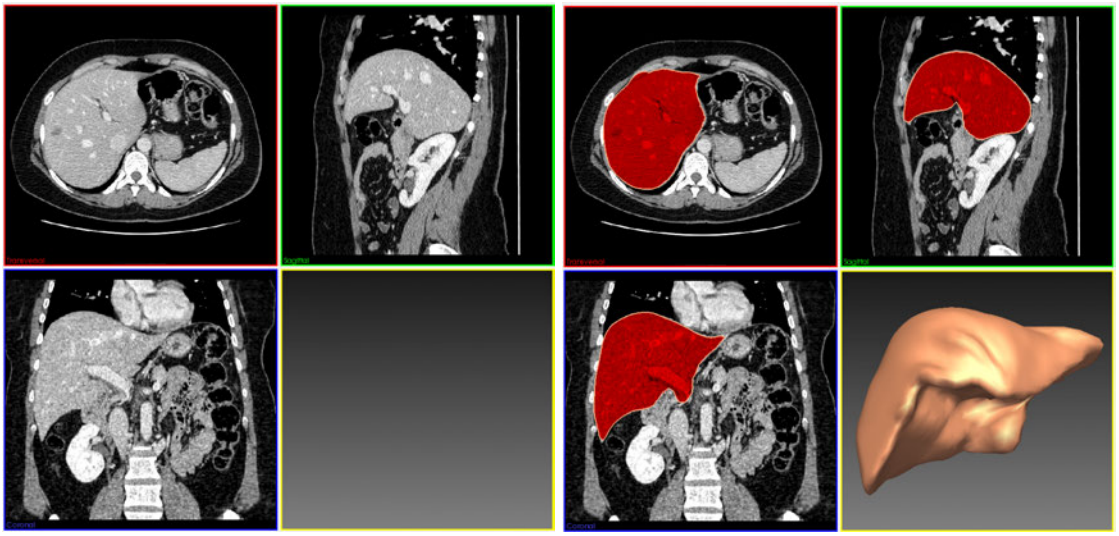


Figure 4.24.: Left: 2D views of a CT dataset. Right: Fully automatic organ segmentation.

with the planning of typical and atypical resections.

4.6.2.1. Stage 1 - Data processing

First of all, the user has to segment the liver from the CT dataset. This task is carried out fully automatically after pushing a 'Start' button. Figure 4.24 shows the CT dataset before (left) and after (right) organ segmentation.

Next, the liver vessels are segmented. Therefore, the user has to provide a few seed points in the main branches of the liver veins. After pushing another 'Start' button, an initial segmentation is calculated. Figure 4.25 shows the segmented organ with three user provided seed points and the initial segmentation (left). The initial segmentation can be refined with a slider until the results are optimal (Figure 4.25, right).

The segmentation of the vessels usually contains both, portal and hepatic veins. Because of low resolutions, motion artifacts and imperfect segmentation algorithms both vessel systems appear to be connected at several points in the CT data. Thus, the next step deals with the separation of both vessel systems. The user has to manually mark root nodes of both vessel systems. Edges to be removed are automatically suggested. At any time, the user can unmark edges or select additional ones. Figure 4.26 shows the

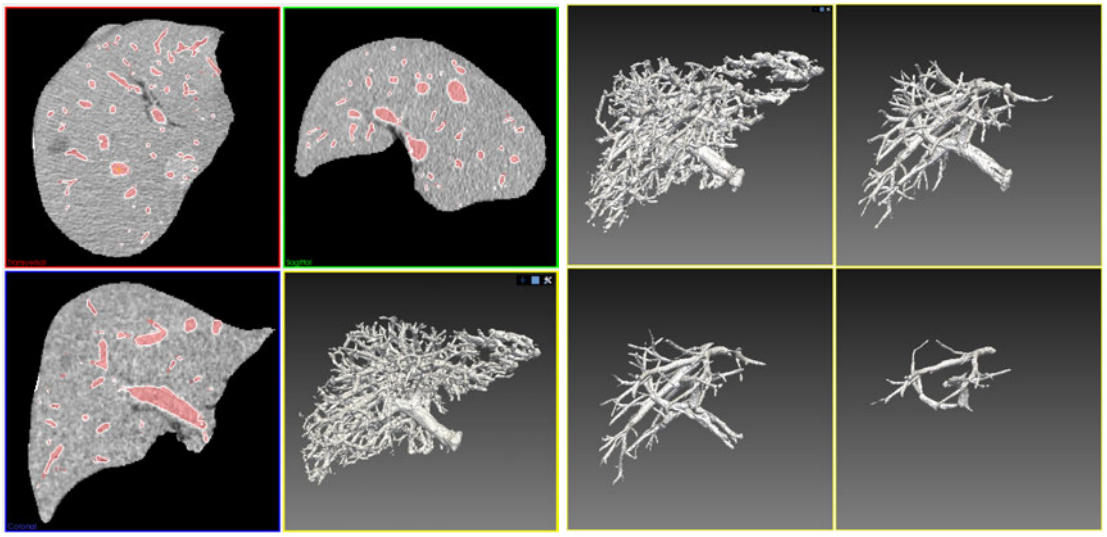


Figure 4.25.: Left: Vessel segmentation results overlapped on 2D slices of cropped liver and 3D visualization. Right: Possible refinements to the initial segmentation. Image taken from [DOL12a].

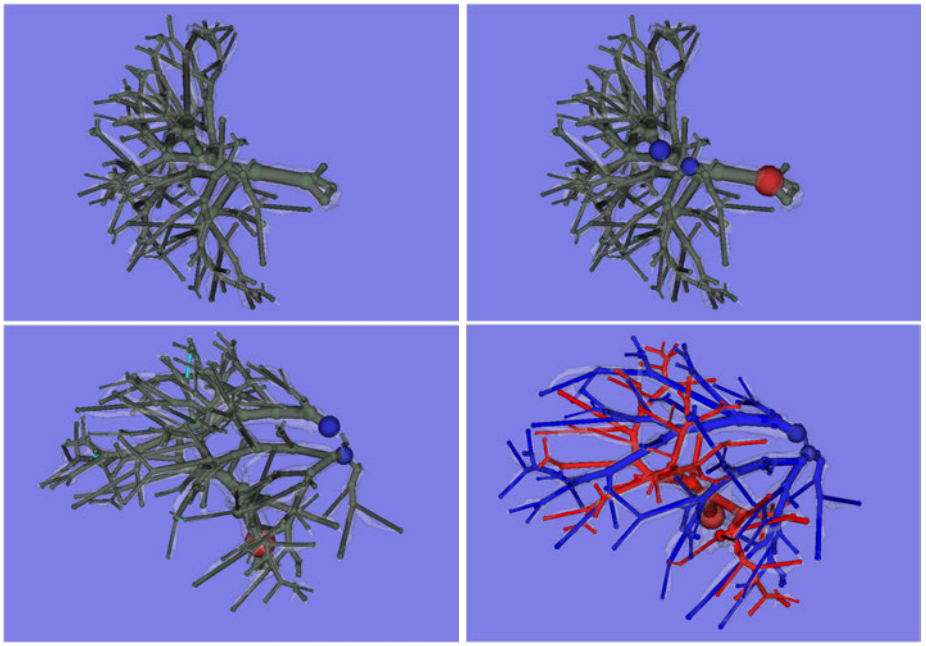


Figure 4.26.: Separation of portal and hepatic veins. From top left to bottom right: Interconnected portal and hepatic vessels, selected root nodes of portal and hepatic vessel trees, automatically suggested edges to be removed, separation result.

main steps of separating portal and hepatic vein. From left to right: Graph representation of the segmented vessels, selected root nodes of portal (red) and hepatic (blue) veins, automatically suggested edges and separation results.

Finally, tumors are marked or segmented. To mark a tumor, the center position of the tumor has to be provided by setting a seed point in one of the 2D slices. A sphere is

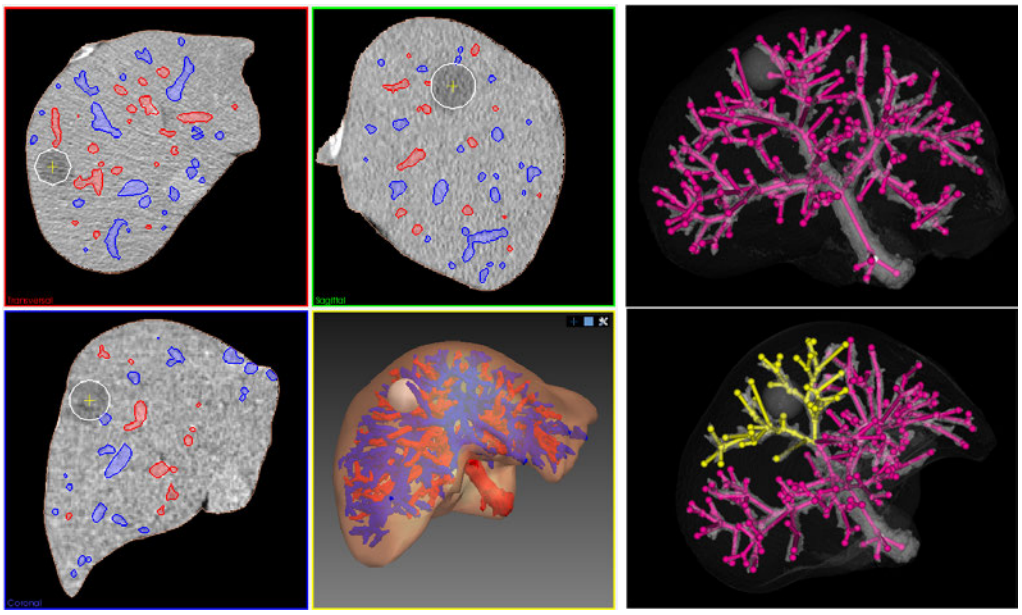


Figure 4.27.: Left: Marking of tumors and overlapped visualization of portal and hepatic veins in 2D views and visualization in 3D. Right top: Visualization of the portal vein with an overlapped abstract representation that is used for interaction. Right bottom: After clicking a node, the corresponding sub-tree is automatically annotated. Image taken from [DOL12a].

then drawn around it. With a slider, the size of the sphere can be adjusted. At any time, the user can grab the sphere and move it around to correct the position. Please note that this is not a real tumor segmentation, but just a marker to visualize the position and approximate size of a tumor. Figure 4.27 shows transversal, sagittal and coronal views in 2D and a 3D view of the organ, vessels and tumor. To segment a tumor, the user has to provide information about foreground and background structures. This is done by drawing lines in arbitrary slices of the dataset in two different colors. One color represents parts of the tumor (foreground), while the other color represents parts that do not belong to the tumor (background). The proposed graph-cuts method is then applied to calculate an initial segmentation. If the results are not satisfactory, additional lines can be drawn until the result is convincing.

4.6.2.2. Stage 2 - Typical resections

At this point, all relevant structures are segmented, namely liver, portal and hepatic veins and tumors. An abstract version of the portal vein is visualized together with the segmented vessels and tumors as transparent backgrounds. To simulate clamping, the user can select nodes or edges. The corresponding sub-tree is then annotated with a new color to indicate visually which parts of the vessels are affected. Figure 4.27 shows the initial visualization (right top) and an annotated sub-tree in yellow (right bottom).

If the annotation is completed, the color change on the liver surface will be calculated. Figure 4.28 shows the results. On the left side, the results are shown in transversal, sagittal and coronal views. These results can also be overlapped on the original CT data

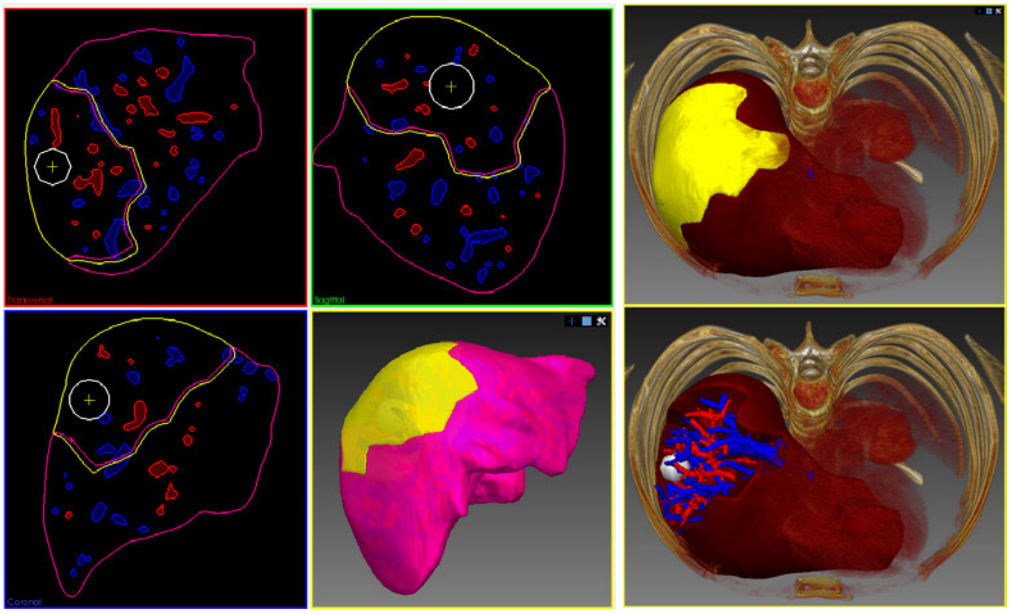
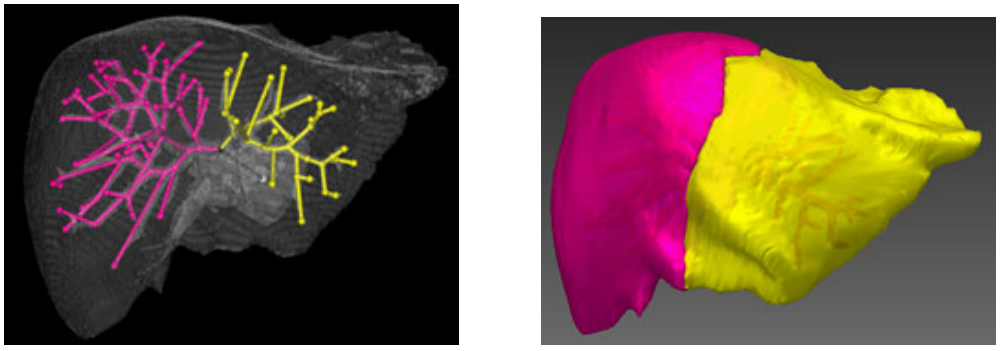


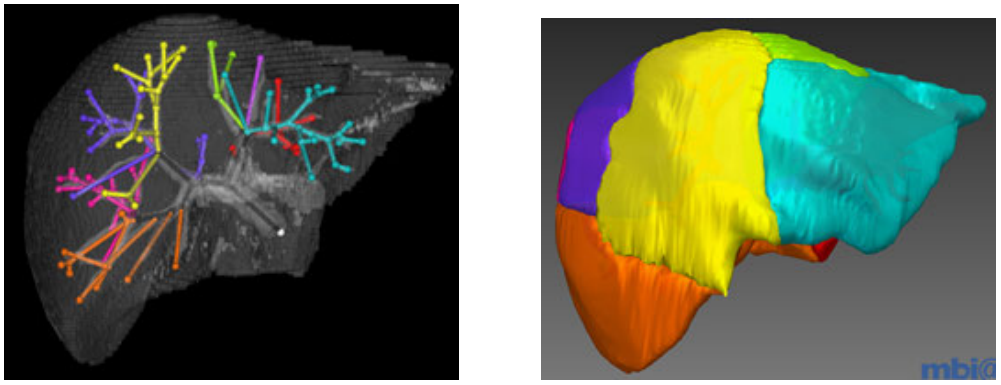
Figure 4.28.: Left: Results in 2D and 3D. The yellow area visualizes the borders on the liver surface when the portal vein is clamped at the given position. Right: Visualization of 3D structures combined with volume rendering of surrounding structures. Image taken from [DOL12a].



(a) (b)

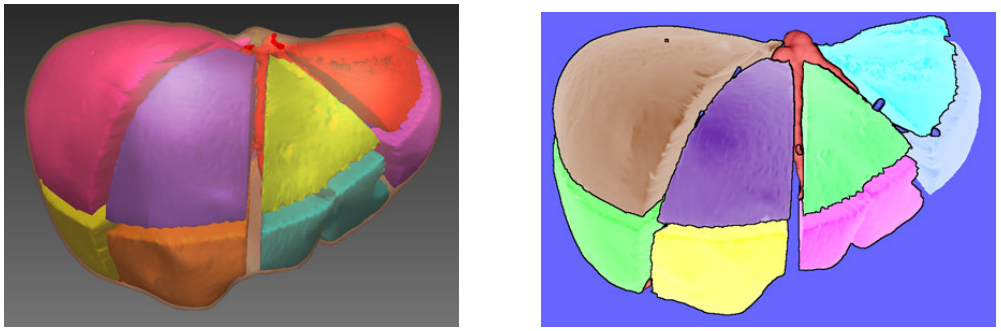
Figure 4.29.: Patient specific segmentation of the liver in left and right. (a) Annotated vessels. (b) Approximation of the corresponding liver segments.

(not shown). On the right side, the results are visualized in the context of surrounding structures using a mix between 3D visualization of surfaces and volume rendering of surrounding structures. Figure 4.29 shows the patient specific segmentation of the liver in left and right. Figure 4.30 shows patient specific approximation of the liver segments as defined by Couinaud. Finally, Figure 4.31 shows results of the more traditional planes-based approach using surface and illustrative rendering with cool to warm tone shading.



(a) (b)

Figure 4.30.: Patient specific approximation of the liver segments as defined by Couinaud. (a) Annotated vessels. (b) Approximation of the corresponding segments.



(a) (b)

Figure 4.31.: Results of the more traditional planes-based approach. (a) Surface rendering (b) Illustrative rendering with cool to warm shading.

4.6.3. Planning - Risk assessment

The proposed methodology is demonstrated retrospectively by two cases where the tumor was only visible in the arterial phase, but not in the venous phase. Thus, without a deformable registration, these cases could not be handled with computer support. Two clinically acquired contrast enhanced multiphase CT datasets were used as input to the system. Table 4.4 summarizes the parameters of the CT datasets which were acquired by a GE Medical Systems LightSpeed Ultra. Both patients did not undergo an open surgery, but were subject to RF (Radio-Frequency) ablation.

Table 4.5 contains the processing times of various steps for both datasets. The processing time of the registration includes the time for organ segmentation, which is approximately 2 minutes. The times for vessel extraction includes segmentation, thinning, graph creation and interactive graph separation. The reported times for risk analysis are for the execution of the algorithm presented in Algorithm 1 for portal venous and hepatic territories. The registration accuracy was evaluated as follows [ESH*11]. For the lower liver part the mean distance between boundaries after registration was 4.5 mm, for the middle liver part 2.3 mm and for the upper liver part 2.2 mm. For the organ segmentation, an average surface distance to a ground-truth segmentation of 1.6 mm was

Dataset	Resolution	Spacing
DS1_Arterial	512x512x274	0.7x0.7x1.25
DS1_Portal	512x512x385	0.7x0.7x1.25
DS2_Arterial	512x512x170	0.64x0.64x1.25
DS2_Portal	512x512x356	0.64x0.64x1.25

Table 4.4.: Parameters of used datasets.

	DS1 [min]	DS2 [min]
Registration	8:52	7:57
Tumor	0:36	0:54
Organ	2:20	2:00
Vessels	7:30	5:00
Risk analysis	0:06	0:08
Summary (w/o organ)	17:04	13:59

Table 4.5.: Processing times.

reported [EKSW10].

Figure 4.32 shows that the tumor is only visible in the arterial phase of both datasets (red contours), but can be fused with the portal phase after deformable registration. In the arterial phase of dataset 2 vessels are hardly visible, but clearly enhanced in the portal venous phase.

Figure 4.33 shows a 3D visualization of the segmented structures from both, arterial and venous phases of dataset 2. Using 3D visualization, location and orientation of the tumor and vessel structures can be analyzed.

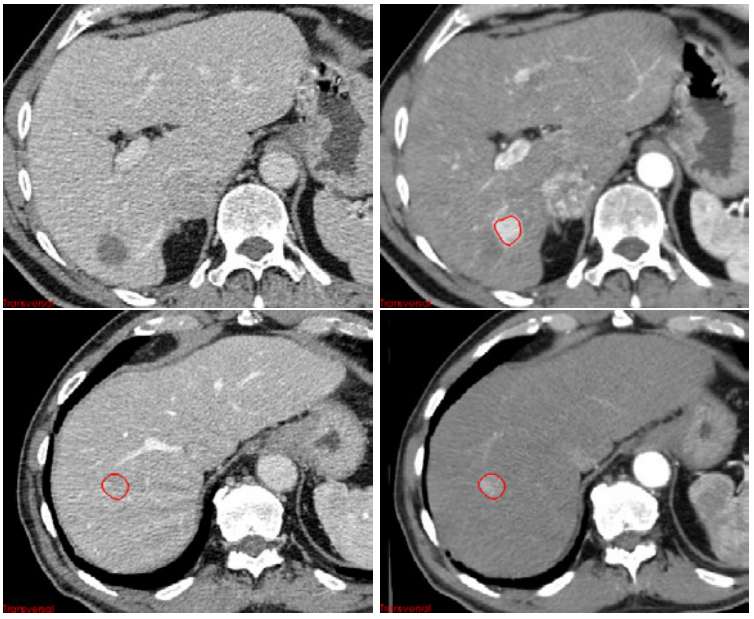


Figure 4.32.: The tumor is only visible in the arterial phase of both CT datasets. Top row: Dataset 1. Bottom row: Dataset 2. Left column: Venous phase. Right column: Arterial phase. Image taken from [DEOLW12].

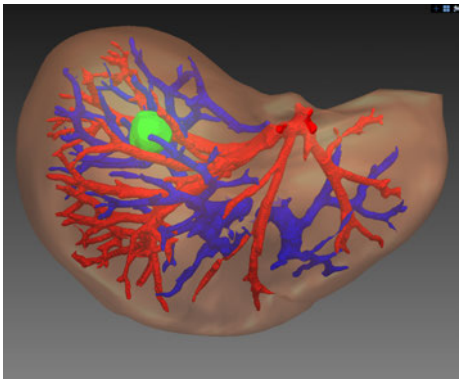


Figure 4.33.: 3D visualization of the segmented structures of Dataset 2. The tumor is segmented in the arterial phase, the vessels in the portal venous phase. Image taken from [DEOLW12].

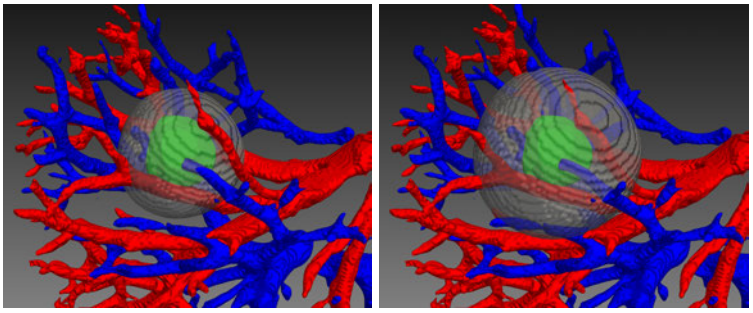


Figure 4.34.: Safety Margin around the tumor in Dataset 2. Left: 1 cm. Right: 1.5 cm. Image taken from [DEOLW12].

Figure 4.34 shows exemplarily two safety margins of 1.0 cm and 1.5 cm respectively around the tumor. Now it can already be visually inspected which vessels would be cut during a resection if this safety margin is closely followed.

Figure 4.35 shows portal venous supply territories (red) and hepatic venous drain territories (blue) for both safety margins. While hepatic venous territories remained constant, the portal venous territory increased when the safety margin was slightly increased. Compared to the size of the tumor, the affected territories are quite big, even if a minimum safety margin of 1.0 cm is followed. It seems reasonable to use RF ablation as the treatment option of choice.

4.7. Discussion

In this chapter, a planning system for typical and atypical resections was developed. Several open source and commercial toolkits were used to realize this application. The basis framework is extended by several plugins which together form a processing pipeline from vessel extraction to tumor segmentation to planning and risk assessment. It provides 2D and 3D visualizations, for example to overlap delineated portal and hepatic veins to the 2D CT slices, and a 3D visualization of the results for better orientation.

For tumor segmentation, a graph-cuts method to automatically identify a ROI in the

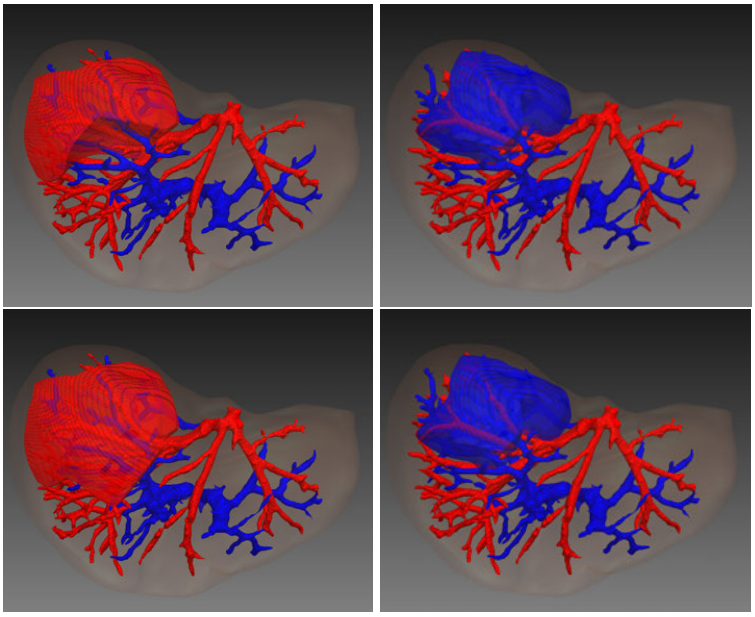


Figure 4.35.: Visualization of the affected territories in dataset 2. Top row: With Safety Margin 1 cm. Bottom row: With Safety Margin 1.5 cm. Left column: Portal venous territories. Right column: Hepatic venous territories. Image taken from [DEOLW12].

original image to reduce graph complexity was presented. Therefore, a coarse resolution image was used to determine a ROI that is used as initialization to construct a reduced graph for final segmentation in the original image. The process of determining a ROI is completely transparent to the user. Experiments with seven tumors showed that the average overlap was 80%, while the average surface distance was 0.73 mm and the average maximum surface distance 5.31 mm. Memory consumption was greatly reduced using the proposed method. For planning of typical liver resections, patient specific liver segments can be calculated. Furthermore, a more traditional approach with rigid planes was developed to determine the liver segments.

Another plugin was presented that can utilize multiple phases of a liver CT dataset in order to assess the risk of atypical resections. Comparable systems lack a multiphase registration module and thus cannot be used if information are spread across CT phases. It was shown by the example of two clinical cases, that this enabled the fusion of complementary information into a 3D representation of the patient anatomy. This information was used to perform a retrospective risk analysis which showed the feasibility of the presented approach. The runtimes of single processing steps were reported. Approximately 15 - 20 minutes are necessary to perform all steps until a risk analysis can be performed.

Preliminary versions of the presented tools have been evaluated by clinical partners at the Universitätsklinikum Frankfurt am Main, Hannover Medical School, and Erasmus Medical Center Rotterdam. Comments were quite positive. The delineation of portal and hepatic veins with different colors was very useful and 3D visualization helped to assess better the position of anatomical structures and pathologies. These results are in line with results found by other groups, e.g. [BTB*10].

5. Conclusions and future work

5.1. Conclusion

Vessel segmentation is a challenging problem that has to deal with acquisition-dependent problems such as noise, contrast, spatial resolution, and artifacts. Furthermore, blood vessel specific characteristics like high variability of size and curvature result in additional difficulties for segmentation algorithms. The liver, in particular, has densely distributed supply and drain vessel systems, and because of partial volume effects and motion artifacts, they seem to be connected at some points. The separation of both vessel systems is a desirable goal to enable a delineated visualization. Furthermore, it enables applications like automated risk analysis and segment approximation.

In this thesis, a vessel extraction methodology was presented that overcomes several shortcomings of existing methods. A comparison of several available vessel enhancement methods revealed that none of them is the clear winner regarding the enhancement of liver vessels. Therefore, a new method for vessel enhancement was proposed. It uses the Laplacian to significantly reduce eigenvalue computations. Furthermore, the scale space representation of vessels is utilized to enhance vessels towards their center. Using the proposed method, nearby vessels are better delineated and segmentation becomes significantly more robust to noise. To segment the liver vessels, a wavefront propagation technique is combined with the proposed vessel enhancement method. It imposes a threshold on the maximum number of voxels that the propagating front must contain to prevent leaks into neighboring areas. Segmentation defects occur at locations where the 'tubular structure' assumption is violated. Therefore, a cascaded voting filter is proposed, which repairs these defects without overly affecting the shape of the vessels. Afterwards, a three-stage process to transform a skeleton into a graph is proposed. It consists of a new skeleton classification scheme, skeleton traversal using a deterministic finite automaton and the decomposition of vessel skeletons into sub-branches for efficient further processing. It does not require a preselected root voxel to produce correctly connected sub-branches. In the last step of this pipeline, the separation of interconnected hepatic veins takes place. It deals with the problem of existing methods that segmentations of vessels do not necessarily follow a clear model assumption. As a matter of fact, a method was developed that takes different violations of model assumptions into account. However, experiments have shown that ambiguous situations can occur which led to wrongly detected edges. In future work, this should be addressed to achieve a fully automatic separation.

The current pipeline is targeted towards interactive use. The results for each stage are shown to the user, who can influence the results using a graphical user interface. This has the advantage that each stage gets the best possible data as input. However, at the same time this is also a disadvantage of the proposed method. To increase its use further,

research should be directed towards the automation of each stage with the ultimate goal to have a fully automatic pipeline. This would allow to use the idle time between data acquisition and human processing of the data more efficiently. The system would then only present the final result of the processing pipeline to the user. Care must be taken to include efficient interaction mechanisms in case something goes wrong. In this case, the user must be able to correct problems at the stage from where the problem originated.

One problem when reading about different vessel segmentation algorithms is their comparability. Each group is using different data, different metrics, expert segmentations and so on. A direct quantitative comparison is impossible. In this thesis, an evaluation framework is proposed, which allows for a standardized validation of segmentation methods. This is important as it allows for a consistent and comparable validation of segmentation methods. Up to date, the user interface is limited to local access. To further increase its usefulness and deploy it in a large-scale, it should be connected to the internet with a web-based interface, which provides possibilities for uploading and downloading files and automatic generation of rankings and evaluation reports.

The aforementioned method was used in an application for interventional planning of typical and atypical liver resections. In this application, a multiresolution method based on graph cuts is used to segment liver tumors. It automatically identifies a region of interest (ROI) in a coarse grid and is used in a fine grid to segment the desired object. Typical resections can be planned by segmenting the liver based on vessel branching or by placing planes within the dataset. Atypical resection can be planned by means of risk analysis. Hereby, it is evaluated which vessels are concerned if the surgeon keeps a specific safety margin around the tumor. The application is build around open source and commercial toolkits, where each module is implemented as plugin. Each plugin can be replaced or additional plugins added. Currently, this application does not provide tools to manually perform a virtual (arbitrary) resection. This, however, would be a useful addition to simulate the outcome of the intervention. Furthermore, it would enhance and complement the risk analysis module.

5.2. Future work

Planning of resections is only one side of the coin. In future work, it is important to bring planning results into the operating room by means of a surgical navigation system. Therefore, a connection between preoperative and intraoperative data has to be established and appropriately presented to the user. The fusion of information is usually done by registering preoperative planning data with an intraoperative modality such as ultrasound. Here, vessels play an important role. They are the only visible landmarks which can be used to establish a correspondence between both modalities, for example by means of tree-to-tree or graph-to-graph matching followed by a deformable registration. However, the liver, being a soft tissue organ, will be highly deformed during intervention, especially in open surgery when the liver is mobilized. A proof-of-concept was provided by Hassenpflug [Has04]. Several challenges remain unsolved. Among them is the robust segmentation of liver vessels in ultrasound data. Future work should address these challenges and develop applications that can be clinically used.

After a liver resection, the question of how good (with respect to the planning) it was performed might arise. The problem of outcome validation can be reduced to a registration problem of preoperative and postoperative data. The challenge is to develop deformable registration algorithms that are able to deal with resected livers in postoperative data, e.g. when one or more pieces are missing. Current registration algorithms reach their limits in this case. As in the navigation challenge, the vessels in both datasets might be of great importance to establish correspondences and to detect resected regions.

Both scenarios require deformable registration algorithms, which utilize correspondences between vessels of both volumes. Various methods have been proposed to register liver volumes based on vessels as landmarks (e.g. [LWRS09,Has04]). They are based on branching points of the vessel trees as anatomical landmarks inherently available in the liver. In case of the liver, early research results used separated hepatic veins as input to a tree matching algorithm (e.g. [Has04, OLD11a]) to establish correspondences. However, these methods require the separation of both veins as an intermediate step. For this reason, current research focuses on graph matching to avoid this step [OLD11c]. The rough idea is sketched as follows. After the segmentation of the liver vessels, a skeleton is created. The skeleton is then transformed into a formal graph representation where the root, the leaves and branching points are represented by attributed nodes. The formal graph representations serve as input to a graph matching algorithm whose task is to automatically find pair-wise correspondences between two vessel trees. The matching is then used to deform one dataset to match the other. The quality of the matching directly influences the result of the registration algorithm. However, to date, no existing automated tree-matching algorithm is able to correctly match all available nodes and, in addition, has been sufficiently tested. Charnoz et al. [CAM*05], for example, reported an algorithm able to match 90-95% of the nodes, while the algorithm by Lohe et al. [LKZ*08] could match 80-90% of the nodes. Metzen et al. [MKS*07] reported that their algorithm produced no false matching. However, they evaluated their algorithm only with one liver dataset and compared only a few matchings that were manually selected as ground truth. Their algorithm found only 50% of those matches. The use of approximating graph-matching methods can be problematic, because wrongly matched correspondences may lead to contortions of the interpolation [Has04]. The reasons for false matchings are mainly due to partial volume effects, inaccurate segmentation algorithms and low spatial resolutions (especially when ultrasound is being used). Thus, the extracted formal trees to be matched are defective and incomplete. This leads to the need of efficient and intuitive interaction mechanisms. Lange et al. noted that the interactive determination of landmarks is tedious and time-consuming [LWRS09]. They motivated the need for an efficient interaction mechanism based on the fact that usually only about 5-6 and rarely up to 10 branching points can be selected during the available time in the operating room [LPH*09]. They proposed a method which visualizes the vessels as surfaces; when the surface near a branch is clicked by the user, the nearest branching point is chosen automatically. Published works in this area concentrated either on completely manual or fully automatic landmark placement. Drechsler et al. [DOLCE10] proposed a combination between completely manual landmark placement and fully automatic tree matching by introducing an efficient interactive component. It provides several visualization features that can be used to support automatic tree matching algorithms with important pre-selected

correspondences or after an automatic matching to manually correct wrongly matched or unmatched nodes. However, while some user interaction is acceptable for the outcome validation scenario, it is still not clear how much interaction is acceptable during surgical navigation.

A. Own Publications, Talks and Presentations

A.1. Publications

A.1.1. Journals

1. DRECHSLER K., OYARZUN LAURA C.: Closing the gap: From planning to intervention in cardiology. *Communications in Computer and Information Science 229* (2011), 187–200

2. OYARZUN LAURA C., DRECHSLER K.: Computer assisted matching of anatomical vessel trees. *Computers & Graphics 35*, 2 (2011), 299–311

3. CHEN Y., DRECHSLER K., OYARZUN LAURA C., ZHAO W., LIU Y.: A graph description and analysis framework of liver vascular system. *Journal of Computational Information Systems 7*, 6 (2011), 1998–2005

4. STALLKAMP J., DRECHSLER K., BERGEN T., KALTENBACHER D., BURISCH M., KAGE A., MÜNZENMAYER C., SAKAS G., WERNER N., WECHSLER A., WINTER C., WITTENBERG T.: Ein neues konzept eines integrierten assistenzsystems für die navigierte weichteilchirurgie. *Endoskopie heute 24*, 1 (2011), P10

5. DRECHSLER K., OYARZUN LAURA C., CHEN Y., ERDT M.: Semi-automatic anatomical tree matching for landmark-based elastic registration of liver volumes. *Journal of Healthcare Engineering 1*, 1 (2010), 101–123

6. STALLKAMP J., DRECHSLER K., BERGEN T., KALTENBACHER D., BURISCH M., KAGE A., MÜNZENMAYER C., SAKAS G., WERNER N., WECHSLER A., WINTER C., WITTENBERG T.: Whole'o'hand - a holistic intervention and interaction system: A novel concept for closed-loop liver surgery. *Biomedizinische Technik / Biomedical Engineering 55*, Suppl. 1 (2010), 27–31

7. BOBDA C., CHENG K., MÜHLBAUER F., DRECHSLER K., SCHULTE J., MURR D., TANOUGAST C.: Enabling self-organization in embedded systems with reconfigurable hardware. *International Journal of Reconfigurable Computing 2009* (2009), Article ID 161458, 9 pages

A.1.2. Conferences & Workshops

1. DRECHSLER K., OYARZUN LAURA C., WESARG S.: Hepatic vein segmentation using wavefront propagation and multiscale vessel enhancement. In *Proceedings of SPIE Medical Imaging* (2013) (accepted)

2. KEIL M., LAURA C. O., DRECHSLER K., WESARG S.: Ct and ultrasound liver vessel registration based on combined segmentation of b-mode and color flow. In *Proceedings of the 3rd Eurographics Workshop on Visual Computing for Biology and Medicine (VCBM)* (2012). Accepted for publication

3. OYARZUN LAURA C., BUENO PLAZA P., DRECHSLER K., WESARG S.: Comparison between rigid registration methods in four clinical scenarios. In *Proceedings of the International MICCAI Workshop on Clinical Image-based Procedures: From Planning to Intervention* (2012), LNCS, Springer. Accepted for publication

4. DRECHSLER K., ERDT M., OYARZUN LAURA C., WESARG S.: Multiphase risk assessment of atypical liver resections. In *Proceedings of the 25th IEEE International Symposium on Computer-Based Medical Systems (CBMS)* (2012), pp. 1–4

5. DRECHSLER K., OYARZUN LAURA CRISTINA ANS WESARG S.: Interventional planning of liver resections: An overview. In *Proceedings of the 34th Annual Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)* (2012), pp. 3744–3747

6. DRECHSLER K., OYARZUN LAURA C.: Simulation of portal vein clamping and the impact of safety margins for liver tumor resections. In *Abdominal Imaging. Computational and Clinical Applications* (2012), pp. 51–59

7. ERDT M., OYARZUN LAURA C., DRECHSLER K., DE BENI S., SOLBIATI L.: Improving diagnosis and intervention: A complete approach for registration of liver ct data. In *Abdominal Imaging. Computational and Clinical Applications* (2012), pp. 108–115

8. OYARZUN LAURA C., DRECHSLER K., ERDT M., KEIL M., NOLL M., BENI S. D., SAKAS G., SOLBIATI L.: Intraoperative registration for liver tumor ablation. In *Abdominal Imaging. Computational and Clinical Applications* (2012), pp. 133–140

9. DRECHSLER K., OELMANN S., OYARZUN LAURA C.: Separation of interconnected hepatic veins. In *Proceedings of the 24th IEEE International Symposium on Computer-Based Medical Systems (CBMS)* (2011), pp. 1–6

10. DRECHSLER K., STROSCHE M., OYARZUN LAURA C.: Automatic roi identification for fast liver tumor segmentation using graph-cuts. In *Proceedings of SPIE Medical Imaging* (2011)

11. CHEN Y., DRECHSLER K., ZHAO W., OYARZUN LAURA C.: A thinning-based liver vessel skeletonization method. In *IEEE International Conference on Internet Computing Information Services (ICICIS)* (2011), pp. 152–155

12. CHEN Y., DRECHSLER K., ZHAO W., OYARZUN LAURA C.: Liver vessel tree generation based on skeletonization and graph representation. In *Proceedings of the 5th IEEE International Conference on Bioinformatics and Biomedical Engineering (iCBBE)* (2011), pp. 1–4

13. ERDT M., KIRSCHNER M., DRECHSLER K., WESARG S., HAMMON M., CAVALLARO A.: Automatic pancreas segmentation in contrast enhanced ct data using learned spatial anatomy and texture descriptors. In *Proceedings of the 8th IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI)* (2011)

14. FRIEDL S., DRECHSLER K., OYARZUN LAURA C., KULESCHOW A., KONDRUWEIT M., WITTENBERG T.: A holistic approach using multimodal image-data for the support of cardiac surgery interventions. In *Proceedings of the 25th Int. Congress & Exhibition CARS 2011: Computer Assisted Radiology and Surgery* (2011), vol. 6, Suppl. 1, pp. 235–238

15. OYARZUN LAURA C., DRECHSLER K.: Graph to graph matching: Facing clinical challenges. In *Proceedings of the 24th IEEE International Symposium on Computer-Based Medical Systems (CBMS)* (2011)

16. OYARZUN LAURA C., DRECHSLER K.: Extracting anatomical landmarks of the liver vasculature using tree matching. In *Biomedical Engineering* (2011), vol. 56

17. TAUSCH R., BURISCH M., DRECHSLER K.: Registration of structured-light 3d scan data with ct skin information for patient positioning. In *Biomedical Engineering* (2011), vol. 56

18. DRECHSLER K., OYARZUN LAURA C.: Comparison of vesselness functions for multiscale analysis of the liver vasculature. In *Proceedings of the 10th IEEE International Conference on Information Technology and Applications in Biomedicine (ITAB)* (2010), pp. 1–5

19. DRECHSLER K., OYARZUN LAURA C.: Hierarchical decomposition of vessel skeletons for graph creation and feature extraction. In *Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (2010), pp. 456–461

20. DRECHSLER K., OYARZUN LAURA C.: A novel multiscale integration approach for vessel enhancement. In *Proceedings of the 23rd IEEE International Symposium on Computer-Based Medical Systems (CBMS)* (2010), pp. 92–97

21. DRECHSLER K., OYARZUN LAURA C.: Dimension reduction based on centroids for multimodal anatomical landmark-based 3d/2d registration of coronary angiograms. In *Proceedings of the International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP)* (2010), INSTICC Press

22. CHEN Y., DRECHSLER K., ZHAO W., OYARZUN LAURA C.: A thinning-based liver vessel skeletonization method. In *Proceedings of the 3rd International Conference on Computational Intelligence and Industrial Application* (2010), vol. 2

23. OYARZUN LAURA C., DRECHSLER K.: Efficient globally optimal matching of anatomical trees of the liver. In *Proceedings of the 2nd Eurographics Workshop on Visual Computing for Biology and Medicine (VCBM)* (2010), pp. 75–82

24. DRECHSLER K., OYARZUN LAURA C.: Registration of cardiac ct data with coronary angiograms using digitally reconstructed radiographs. In *Proceedings of the 11th International World Congress on Medical Physics and Biomedical Engineering* (2009), Düssel O., Schlegel W., (Eds.), Springer, pp. 1431–1434

25. DRECHSLER K., OYARZUN LAURA C., SAKAS G.: Towards computer assisted cardiac catheterization - how 3d visualization supports it. In *Proceedings of the International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP)* (2009), INSTICC Press, pp. 159–165

26. CHEN Y., OYARZUN LAURA C., DRECHSLER K.: Generation of a graph representation from threedimensional skeletons of the liver vasculature. In *Proceedings of the 2nd IEEE International Conference on BioMedical Engineering and Informatics (BMEI)* (October 2009), Shi R., Fu W., Wang Y., Wang H., (Eds.), IEEE

27. WITTENBERG T., DRECHSLER K., KALTENBACHER D., FRIEDL S., REIS C., SAKAS G., STALLKAMP J., KONDRUWEIT M.: 'miss heart': Assisting systems for minimal invasive smart suturing in cardiac surgery - a conceptually closed-loop approach. In *Proceedings of the 11th International World Congress on Medical Physics and Biomedical Engineering* (2009), Dössel O., Schlegel W., (Eds.), Springer, pp. 445–448

28. DRECHSLER K., SCHULTE J., MURR D., BOBDA C.: Enabling self-organization in embedded systems with reconfigurable hardware. In *Proceedings of the 4th International Workshop on Reconfigurable Communication Centric System-on-Chips (ReCoSoC)* (2008)

29. DRECHSLER K., SCHULTE J., MURR D., MÜHLBAUER F., BOBDA C.: Self-organization in networked reconfigurable systems for computer vision applications. In *Workshop on Design and Architectures for Signal and Image Processing (DASIP)* (2007)

A.2. Talks

1. Annual Conference of the IEEE Engineering in Medicine & Biology Society (EMBC). *Interventional Planning of Liver Resections: An Overview*. San Diego, California, USA. **Invited lecture**

2. IEEE International Symposium on Computer-Based Medical Systems (CBMS). *Multiphase Risk Assessment of Atypical Liver Resections*. Rome, Italy. 2012.

3. MICCAI International Workshop on Computational and Clinical Applications in Abdominal Imaging. *Simulation of Portal Vein Clamping and the Impact of Safety Margins for Liver Tumor Resections*. Toronto, Canada. 2011.

4. IEEE International Symposium on Computer-Based Medical Systems (CBMS). *Separation of Interconnected Hepatic Veins*. Bristol, England. 2011.

5. Symposium der DGBMT: "Krankenhaus der Zukunft". *Closing the Gap: From Rigid to Deformable to Closed Loop Surgery*. Munich, Germany. 2011. **Invited talk**

6. International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP). *Dimension Reduction Based On Centroids For Multimodal Anatomical Landmark-based 3d/2d Registration Of Coronary Angiograms*. Lisbon, Portugal. 2010.

7. IEEE International Symposium on Computer-Based Medical Systems (CBMS). *A Novel Multiscale Integration Approach for Vessel Enhancement*. Perth, Australia. 2010.

8. IEEE International Conference on Bioinformatics and Biomedicine (BIBM). *Hierarchical Decomposition of Vessel Skeletons for Graph Creation and Feature Extraction*. Hong Kong, China. 2010.

9. IEEE International Conference on Information Technology and Applications in Biomedicine (ITAB). *Comparison of vesselness functions for multiscale analysis of the liver vasculature*. Corfu, Greece. 2010.

10. IEEE International Conference on BioMedical Engineering and Informatics (BMEI). *Generation of a Graph Representation from Threedimensional Skeletons of the Liver Vasculature*. Beijing, China. 2009.

11. International World Congress on Medical Physics and Biomedical Engineering. *Registration of Cardiac CT Data with Coronary Angiograms using Digitally Reconstructed Radiographs*. Munich, Germany. 2009.

12. International Embedded Systems Symposium (IESS). *Utilizing Reconfigurable Hardware to optimize Workflows in Networked Nodes*. Irvine, California, USA. 2007.

13. International Embedded Systems Symposium (IESS). *Embedded Vertex Shader in FPGA*. Irvine, California, USA. 2007.

14. Winterschool: "Self-Organisation in Embedded Systems". *Self-Organization in Embedded Systems by the example of a Distributed Intelligent Camera System*. Schloss Dagstuhl, Wadern, Germany. 2006.

B. Supervising Activities

The following list summarizes the student bachelor, diploma and master thesis supervised by the author. The results of these works were partially used as an input into the thesis.

B.1. Diploma and Master Thesis

1. KREITZ C.: *Segmentvisualisierung der Leber für die intraoperative Navigation*. Master's thesis, Technische Universität Darmstadt, 2009
2. AKRAM S.: *Coronary Tree Segmentation Using an Enhanced Corkscrew Algorithm*. Master's thesis, Technische Universität Darmstadt, 2009

B.2. Bachelor Thesis

1. MEIXNER S.: *Konzeption und Realisierung einer Testumgebung zum standardisierten Evaluieren von Gefäßsegmentierungs-Algorithmen*. Bachelor's thesis, Hochschule Darmstadt, 2012
2. OELMANN S.: *Konzeption und Realisierung eines Verfahrens zur automatischen Erkennung von fehlerhaften Verbindungsstellen in einer Graphenrepräsentation des Gefäßsystems der Leber*. Bachelor's thesis, Hochschule Darmstadt, 2011
3. STROSCHE M.: *Interaktive Tumor-Segmentierung in medizinischen Volumendaten*. Bachelor's thesis, Technische Universität Darmstadt, 2010
4. SCHULTE J.: *Ein Monitor-Service für Selbstorganisierende Systeme*. Bachelor's thesis, Technische Universität Kaiserslautern, 2007

C. Curriculum Vitae

Personal Data

Name	Klaus Drechsler
Birth date, place	16. April 1976 in Frankfurt am Main
Nationality	German

Education

2003	**Technische Universität Darmstadt** Graduation in Computer Science (Dipl.-Inform.)
1997-2003	**Technische Universität Darmstadt** Study of Computer Science
1996-1997	**Frankfurter Verband für Alten- u. Behindertenhilfe e.V.** Alternative civilian service
1996	**Carl-Schurz Gymnasium, Frankfurt am Main** Abitur

Work Experience

2007-now	**Fraunhofer-Institute for Computer Graphics Research IGD** Researcher
2006-2007	**Technische Universität Kaiserslautern** Researcher
2004-2006	**PEAK-System Technik GmbH, Darmstadt** Software-Developer
1999-2003	**Center for Computer Graphics Research (ZGDV e.V.), Darmstadt** Student Research Assistant

Bibliography

[AB02] AYLWARD S. R., BULLIT E.: Initialization, noise, singularities, and scale in height ridge traversal for tubular object centerline extraction. *IEEE Transactions on Medical Imaging 21* (2002), 61–75.

[Akr09] AKRAM S.: *Coronary Tree Segmentation Using an Enhanced Corkscrew Algorithm*. Master's thesis, Technische Universität Darmstadt, 2009.

[AMR*10] ADEL M., MOUSSAOUI A., RASIGNI M., BOURENNANE S., HAMAMI L.: Statistical-based tracking technique for linear structures detection: Application to vessel segmentation in medical images. *IEEE Signal Processing Letters 17*, 6 (june 2010), 555–558.

[BB08] BAUER C., BISCHOF H.: A novel approach for detection of tubular objects and its application to medical image analysis. In *Proceedings of the 30th DAGM Symposium on Pattern Recognition* (2008), LNCS, Springer.

[BBB*07] BEICHEL R., BAUER C., BORNIK A., SORANTIN E., BISCHOF H.: Liver segmentation in ct data: A segmentation refinement approach. In *Proceedings of the MICCAI Workshop 3-D Segmentatation in the Clinic: A Grand Challenge* (2007), pp. 235–245.

[BC11] BENMANSOUR F., COHEN L.: Tubular structure segmentation based on minimal path method and anisotropic enhancement. *International Journal of Computer Vision 92* (2011), 192–210. 10.1007/s11263-010-0331-0.

[BCM*09] BOBDA C., CHENG K., MÜHLBAUER F., DRECHSLER K., SCHULTE J., MURR D., TANOUGAST C.: Enabling self-organization in embedded systems with reconfigurable hardware. *International Journal of Reconfigurable Computing 2009* (2009), Article ID 161458, 9 pages.

[BJ01] BOYKOV Y. Y., JOLLY M.-P.: Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In *International Conference on Computer Vision* (Los Alamitos, CA, USA, 2001), vol. 1, IEEE Computer Society, p. 105.

[BK04] BOYKOV Y., KOLMOGOROV V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence 26*, 9 (2004), 1124–1137.

[BKBP08] BOCK S., KÜHNEL C., BOSKAMP T., PEITGEN H.-O.: Robust vessel segmentation. In *Proceedings of SPIE Medical Imaging* (2008).

[Blu67] BLUM H.: A Transformation for Extracting New Descriptors of Shape. *Models for the Perception of Speech and Visual Form* (1967), 362–380.

[BMK10] BASHAR M., MORI K., KOBAYASHI T.: Automatic segmentation of abdominal blood vessels from contrasted x-ray ct images. In *Electrical and Computer Engineering (ICECE), 2010 International Conference on* (dec. 2010), pp. 187 –190.

[BPJ*04] BEICHEL R., POCK T., JANKO C., ZOTTER R. B., REITINGER B., BORNIK A., PALAGYI K., SORANTIN E., WERKGARTNER G., BISCHOF H., SONKA M.: Liver segment approximation in ct data for surgical resection planning. In *SPIE Medical Imaging 2004: Image Processing* (2004), Fitzpatrick J. M., Sonka M., (Eds.), vol. 5370, SPIE, pp. 1435–1446.

[BPS*10] BAUER C., POCK T., SORANTIN E., BISCHOF H., BEICHEL R.: Segmentation of interwoven 3d tubular tree structures utilizing shape priors and graph cuts. *Medical Image Analysis 14*, 2 (2010), 172–184.

[BTB*10] BEERMANN J., TETZLAFF R., BRUCKNER T., SCHÖEBINGER M., MÜLLER-STICH B. P., GUTT C. N., MEINZER H.-P., KADMON M., FISCHER L.: Three-dimensional visualisation improves understanding of surgical liver anatomy. *Medical Education 44*, 9 (Sep 2010), 936–940.

[BvAS*07] BENNINK H. E., VAN ASSEN H. C., STREEKSTRA G. J., TER WEE R., SPAAN J. A. E., TER HAAR ROMENY B. M.: A novel 3d multi-scale lineness filter for vessel detection. In *MICCAI* (2007), vol. 4792 of *LNCS*, Springer, pp. 436–443.

[BWKR03] BEZY-WENDLING J., KRETOWSKI M., ROLLAND Y.: Hepatic tumor enhancement in computed tomography: combined models of liver perfusion and dynamic imaging. *Computers in Biology And Medicine 33* (2003), 77–89.

[CAM*05] CHARNOZ A., AGNUS V., MALANDAIN G., SOLER L., TAJINE M.: Tree matching applied to vascular system. In *Graph-based Representations in Pattern Recognition* (2005), vol. 3434 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 183–192.

[CC06] CAI W., CHUNG A. C. S.: Multi-resolution vessel segmentation using normalized cuts in retinal images. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)* (2006), pp. 928–936.

[CC07] CAMPADELLI P., CASIRAGHI E.: Liver segmentation from ct scans: A survey. In *Applications of Fuzzy Sets Theory*, Masulli F., Mitra S., Pasi G., (Eds.), vol. 4578 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2007, pp. 520–528.

[CCE09] CAMPADELLI P., CASIRAGHI E., ESPOSITO A.: Liver segmentation from computed tomography scans: A survey and a new algorithm. *Arificial Intelligence in Medicine 45*, 2-3 (2009), 185–196.

[CDOL*11] CHEN Y., DRECHSLER K., OYARZUN LAURA C., ZHAO W., LIU Y.: A graph description and analysis framework of liver vascular system. *Journal of Computational Information Systems 7*, 6 (2011), 1998–2005.

[CDZOL10] CHEN Y., DRECHSLER K., ZHAO W., OYARZUN LAURA C.: A thinning-based liver vessel skeletonization method. In *Proceedings of the 3rd International Conference on Computational Intelligence and Industrial Application* (2010), vol. 2.

[CDZOL11a] CHEN Y., DRECHSLER K., ZHAO W., OYARZUN LAURA C.: Liver vessel tree generation based on skeletonization and graph representation. In *Proceedings of the 5th IEEE International Conference on Bioinformatics and Biomedical Engineering (iCBBE)* (2011), pp. 1–4.

[CDZOL11b] CHEN Y., DRECHSLER K., ZHAO W., OYARZUN LAURA C.: A thinning-based liver vessel skeletonization method. In *IEEE International Conference on Internet Computing Information Services (ICICIS)* (2011), pp. 152–155.

[CFD*11] CONVERSANO F., FRANCHINI R., DEMITRI C., MASSOPTIER L., MONTAGNA F., MAFFEZZOLI A., MALVASI A., CASCIARO S.: Hepatic vessel segmentation for 3d planning of liver surgery:: Experimental evaluation of a new fully automatic algorithm. *Academic Radiology In Press, Corrected Proof* (2011), –.

[CGC10] CHOUILLARD E. K., GUMBS A. A., CHERQUI D.: Vascular clamping in liver surgery: physiology, indications and techniques. In *Annals of Surgical Innovation and Research* (2010), vol. 4.

[Cha07] CHANG S.: Extracting skeletons from distance maps. *International Journal of Computer Science and Network Security (IJCSNS) 7*, 7 (2007), 213–219.

[CJDL11] CHEN L., JU Y., DING S., LIU X.: Topological vascular tree segmentation for retinal images using shortest path connection. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)* (sept. 2011), pp. 2137 –2140.

[CKL*11] COLLINS J., KURCZ C., LISLE C., LIU Y., ZUDAIRE E.: Assessment of vascular network segmentation. *International Journal of Image Processing 4* (2011), 584–599.

[CLS03] CHOI W.-P., LAM K.-M., SIU W.-C.: Extraction of the euclidean skeleton based on a connectivity criterion. *Pattern Recogn. 36*, 3 (2003), 721 – 729.

[CLV*10] CHI Y., LIU J., VENKATESH S., HUANG S., ZHOU J., TIAN Q., NOWINSKI W.: Segmentation of liver vasculature from contrast enhanced ct images using context-based voting. *IEEE Transactions on Biomedical Engineering PP*, 99 (2010), 1.

[COLD09] CHEN Y., OYARZUN LAURA C., DRECHSLER K.: Generation of a graph representation from threedimensional skeletons of the liver vasculature. In *Proceedings of the 2nd IEEE International Conference on BioMedical Engineering and Informatics (BMEI)* (October 2009), Shi R., Fu W., Wang Y., Wang H., (Eds.), IEEE.

[CR03] CANERO C., RADEVA P.: Vesselness enhancement diffusion. *Pattern Recognition Letters 24*, 16 (2003), 3141–3151.

[CZP*10] CHEN K., ZHANG Y., POHL K., SYEDA-MAHMOOD T., SONG Z., WONG S.: Coronary artery segmentation using geometric moments based tracking and snake-driven refinement. In *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE* (31 2010-sept. 4 2010), pp. 3133 –3137.

[DEOLW12] DRECHSLER K., ERDT M., OYARZUN LAURA C., WESARG S.: Multiphase risk assessment of atypical liver resections. In *Proceedings of the 25th IEEE International Symposium on Computer-Based Medical Systems (CBMS)* (2012), pp. 1–4.

[DOL09] DRECHSLER K., OYARZUN LAURA C.: Registration of cardiac ct data with coronary angiograms using digitally reconstructed radiographs. In *Proceedings of the 11th International World Congress on Medical Physics and Biomedical Engineering* (2009), Düssel O., Schlegel W., (Eds.), Springer, pp. 1431–1434.

[DOL10a] DRECHSLER K., OYARZUN LAURA C.: Comparison of vesselness functions for multiscale analysis of the liver vasculature. In *Proceedings of the 10th IEEE International Conference on Information Technology and Applications in Biomedicine (ITAB)* (2010), pp. 1–5.

[DOL10b] DRECHSLER K., OYARZUN LAURA C.: Dimension reduction based on centroids for multimodal anatomical landmark-based 3d/2d registration of coronary angiograms. In *Proceedings of the International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP)* (2010), INSTICC Press.

[DOL10c] DRECHSLER K., OYARZUN LAURA C.: Hierarchical decomposition of vessel skeletons for graph creation and feature extraction. In *Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (2010), pp. 456–461.

[DOL10d] DRECHSLER K., OYARZUN LAURA C.: A novel multiscale integration approach for vessel enhancement. In *Proceedings of the 23rd IEEE International Symposium on Computer-Based Medical Systems (CBMS)* (2010), pp. 92–97.

[DOL11] DRECHSLER K., OYARZUN LAURA C.: Closing the gap: From planning to intervention in cardiology. *Communications in Computer and Information Science 229* (2011), 187–200.

[DOL12a] DRECHSLER K., OYARZUN LAURA C.: Simulation of portal vein clamping and the impact of safety margins for liver tumor resections. In *Abdominal Imaging. Computational and Clinical Applications* (2012), pp. 51–59.

[DOL12b] DRECHSLER K., OYARZUN LAURA CRISTINA ANS WESARG S.: Interventional planning of liver resections: An overview. In *Proceedings of the 34th Annual Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)* (2012), pp. 3744–3747.

[DOLCE10] DRECHSLER K., OYARZUN LAURA C., CHEN Y., ERDT M.: Semi-automatic anatomical tree matching for landmark-based elastic registration of liver volumes. *Journal of Healthcare Engineering 1*, 1 (2010), 101–123.

[DOLS09] DRECHSLER K., OYARZUN LAURA C., SAKAS G.: Towards computer assisted cardiac catheterization - how 3d visualization supports it. In *Proceedings of the International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP)* (2009), INSTICC Press, pp. 159–165.

[DOLW13] DRECHSLER K., OYARZUN LAURA C., WESARG S.: Hepatic vein segmentation using wavefront propagation and multiscale vessel enhancement. In *Proceedings of SPIE Medical Imaging* (2013).

[DOOL11] DRECHSLER K., OELMANN S., OYARZUN LAURA C.: Separation of interconnected hepatic veins. In *Proceedings of the 24th IEEE International Symposium on Computer-Based Medical Systems (CBMS)* (2011), pp. 1–6.

[DSM*07] DRECHSLER K., SCHULTE J., MURR D., MÜHLBAUER F., BOBDA C.: Self-organization in networked reconfigurable systems for computer vision applications. In *Workshop on Design and Architectures for Signal and Image Processing (DASIP)* (2007).

[DSMB08] DRECHSLER K., SCHULTE J., MURR D., BOBDA C.: Enabling self-organization in embedded systems with reconfigurable hardware. In *Proceedings of the 4th International Workshop on Reconfigurable Communication Centric System-on-Chips (ReCoSoC)* (2008).

[DSOL11] DRECHSLER K., STROSCHE M., OYARZUN LAURA C.: Automatic roi identification for fast liver tumor segmentation using graph-cuts. In *Proceedings of SPIE Medical Imaging* (2011).

[DZF*10] DEBARBA H., ZANCHET D., FRACARO D., MACIEL A., KALIL A.: Efficient liver surgery planning in 3d based on functional segment classification and volumetric information. In *Engineering in Medicine and Biology Society*

(EMBC), 2010 Annual International Conference of the IEEE (September 2010), pp. 4797–4800.

[ECOP07] ESPONA L., CARREIRA M., ORTEGA M., PENEDO M.: A snake for retinal vessel segmentation. In *Pattern Recognition and Image Analysis*, vol. 4478 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2007, pp. 178–185.

[EKD*11] ERDT M., KIRSCHNER M., DRECHSLER K., WESARG S., HAMMON M., CAVALLARO A.: Automatic pancreas segmentation in contrast enhanced ct data using learned spatial anatomy and texture descriptors. In *Proceedings of the 8th IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI)* (2011).

[EKSW10] ERDT M., KIRSCHNER M., STEGER S., WESARG S.: Fast automatic liver segmentation combining learned shape priors with observed shape deviation. In *Proceedings of the IEEE International Symposium on Computer-Based Medical Systems (CBMS)* (2010), pp. 249–254.

[ELD10] ESNEAULT S., LAFON C., DILLENSEGER J.-L.: Liver vessels segmentation using a hybrid geometrical moments/graph cuts method. *IEEE Transactions on Biomedical Engineering 57*, 2 (February 2010), 276–283.

[EMGF07] EGGER J., MOSTARKIC Z., GROSSKOPF S., FREISLEBEN B.: A fast vessel centerline extraction algorithm for catheter simulation. In *Proceedings of the IEEE International Symposium on Computer-Based Medical Systems (CBMS)* (june 2007), pp. 177 –182.

[EOLD*12] ERDT M., OYARZUN LAURA C., DRECHSLER K., DE BENI S., SOLBIATI L.: Improving diagnosis and intervention: A complete approach for registration of liver ct data. In *Abdominal Imaging. Computational and Clinical Applications* (2012), pp. 108–115.

[ER02] ETZION M., RAPPOPORT A.: Computing voronoi skeletons of a 3-d polyhedron by space subdivision. *Computational Geometry 21* (2002), 87–120.

[ERS08] ERDT M., RASPE M., SÜHLING M.: Automatic hepatic vessel segmentation using graphics hardware. In *Proc. MIAR* (2008), Dohi T., Sakuma I., Liao H., (Eds.), vol. 5128 of *LNCS*, Springer, pp. 403–412.

[ESH*11] ERDT M., SAKAS G., HAMMON M., DE BENI S., SOLBIATI L., CAVALLARO A.: Automatic shape based deformable registration of multiphase contrast enhanced liver ct volumes. In *Medical Imaging 2011: Image Processing. Part One* (2011), Proceedings of SPIE; 7962, The International Society for Optical Engineering (SPIE), SPIE Press, Bellingham, pp. 796212–1–796212–6.

[Fas08] FASEL J. H. D.: Portal venous territories within the human liver: an anatomical reappraisal. *Anat Rec (Hoboken) 291*, 6 (Jun 2008), 636–642.

[FCN*04] FRERICKS B. B., CALDARONE F. C., NASHAN B., SAVELLANO D. H., STAMM G., KIRCHHOFF T. D., SHIN H.-O., SCHENK A., SELLE D., SPINDLER W., KLEMPNAUER J., PEITGEN H.-O., GALANSKI M.: 3d ct modeling of hepatic vessel architecture and volume calculation in living donated liver transplantation. *European Radiology 14*, 2 (2004), 326–333.

[FDOL*11] FRIEDL S., DRECHSLER K., OYARZUN LAURA C., KULESCHOW A., KONDRUWEIT M., WITTENBERG T.: A holistic approach using multimodal image-data for the support of cardiac surgery interventions. In *Proceedings of the 25th Int. Congress & Exhibition CARS 2011: Computer Assisted Radiology and Surgery* (2011), vol. 6, Suppl. 1, pp. 235–238.

[FJS09] FREIMAN M., JOSKOWICZ L., SOSNA J.: A variational method for vessels segmentation: Algorithm and application to liver vessels visualization. In *Proceedings of SPIE - Medical Imaging 2009: Visualization, Image-Guided Procedures, and Modeling* (2009), vol. 7261, p. 72610H.

[FMP10] FASEL J. H. D., MAJNO P. E., PEITGEN H.-O.: Liver segments: an anatomical rationale for explaining inconsistencies with couinaudâĂŹs eight-segment concept. *Surgical and Radiologic Anatomy* (2010).

[FNVV98] FRANGI A. F., NIESSEN W. J., VINCKEN K. L., VIERGEVER M. A.: Multiscale vessel enhancement filtering. In *MICCAI* (1998), vol. 1496 of *LNCS*, Springer.

[FP08] FEI Y., PARK J. W.: Automatic segmentation of liver blood vessels using level set method. In *Proceedings of the International Conference on Audio, Language and Image Processing (ICALIP)* (July 2008), pp. 1718–1720.

[FRH*12] FRAZ M., REMAGNINO P., HOPPE A., UYYANONVARA B., RUDNICKA A., OWEN C., BARMAN S.: Blood vessel segmentation methodologies in retinal images ÃćÂĂÂŞ a survey. *Computer Methods and Programs in Biomedicine*, 0 (2012), –.

[FSK07] FURUKAWA D., SHIMIZU A., KOBATAKE H.: Automatic liver segmentation based on maximum a posterior probability estimation and level set method. In *Proceedings of the MICCAI Workshop 3-D Segmentatation in the Clinic: A Grand Challenge* (2007), pp. 117–124.

[FZSH12] FORUZAN A., ZOROOFI R., SATO Y., HORI M.: A hessian-based filter for vascular segmentation of noisy hepatic ct scans. *International Journal of Computer Assisted Radiology and Surgery 7* (2012), 199–205. 10.1007/s11548-011-0640-y.

[GGD*98] GÖPFERT M. R., GLOMBITZA G.-P., DEMIRIS A. M., LAMADÉ W., MEINZER H.-P.: Trennung von gefäßbäumen in medizinischen schichtbildserien am beispiel der leber. In *Bildverarbeitung für die Medizin* (1998).

[GK97] GROSSKOPF S., KIEBER M.: Analyse der struktur von gefäÃ§bäumen. In *Proceedings des 5. Workshops für Digitale Bildverarbeitung in der Medizin* (1997), pp. 121–126.

[GKL*09] GOMAA A. I., KHAN S. A., LEEN E. L. S., WAKED I., TAYLOR-ROBINSON S. D.: Diagnosis of hepatocellular carcinoma. *World Journal of Gastroenterology 15*, 11 (Mar 2009), 1301–1314.

[GKS*93] GERIG G., KOLLER T., SZÉKELY G., BRECHBÜHLER C., KÜBLER O.: Symbolic description of 3-d structures applied to cerebral vessel tree obtained from mr angiography volume data. In *Proceedings of the 15th International Conference on Information Processing in Medical Imaging (IPMI)* (1993), Barrett H. H., Gmitro A. F., (Eds.), vol. 687 of *Lecture Notes in Computer Science*, Springer, pp. 94–111.

[GLD*99] GLOMBITZA G., LAMADÉ W., DEMIRIS A. M., GÖPFERT M.-R., MAYER A., BAHNER M. L., MEINZER H.-P., RICHTER G., LEHNERT T., HERFARTH C.: Virtual planning of liver resections: image processing, visualization and volumetric evaluation. *International Journal of Medical Informatics 53* (1999), 225–237.

[Has04] HASSENPFLUG P.: *Gefäß basierte Registrierung zur Computer gestützten Navigation in der Leberchirurgie*. PhD thesis, Ruprecht-Karls-Universität, 2004.

[HBT05] HEIJMANS H., BUCKLEY M., TALBOT H.: Path openings and closings. *J. Math. Imaging Vis. 22* (May 2005), 107–119.

[HGP05] HAHN H., GEORG M., PEITGEN H.-O.: Fractal aspects of three-dimensional vascular constructive optimization. In *Fractals in Biology and Medicine*, Losa G. A., Merlini D., Nonnenmacher T. F., Weibel E. R., (Eds.), Mathematics and Biosciences in Interaction. Birkhäuser Basel, 2005, pp. 55–66.

[HJ10] HAMARNEH G., JASSI P.: Vascusynth: Simulating vascular trees for generating volumetric image data with ground-truth segmentation and tree analysis. *Computerized Medical Imaging and Graphics 34*, 8 (December 2010), 605–616.

[HLZ*10] HANSEN C., LINDOW B., ZIDOWITZ S., SCHENK A., PEITGEN H.-O.: Towards automatic generation of resection surfaces for liver surgery planning. In *Proceedings of Computer Assisted Radiology and Surgery (CARS)* (2010), vol. 5, pp. 119–120.

[HM09] HEIMANN T., MEINZER H.-P.: Statistical shape models for 3D medical image segmentation: A review. *Medical Image Analysis 13*, 4 (2009), 543–563.

[HPSP01] HAHN H. K., PREIM B., SELLE D., PEITGEN H. O.: Visualization and interaction techniques for the exploration of vascular structures. In *Proceedings of the IEEE conference on Visualization (VIS)* (2001), pp. 395–402.

[HTKM04a] HEIMANN T., THORN M., KUNERT T., MEINZER H.-P.: Empirische vergleichsmaße für die evaluation von segmentierungsergebnissen. In *Bildverarbeitung für die Medizin (BVM)* (2004).

[HTKM04b] HEIMANN T., THORN M., KUNERT T., MEINZER H.-P.: New methods for leak detection and contour correction in seeded region growing segmentation. In *Proceedings of the International Archives of Photogrammetry and Remote Sensing and Spatial Information Sciences* (2004), vol. XXXV, pp. 317–322.

[HVN08] HOMANN H., VESOM G., NOBLE J.: Vasculature segmentation of ct liver images using graph cuts and graph-based analysis. In *5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI)* (May 2008), pp. 53–56.

[HZH*09] HANSEN C., ZIDOWITZ S., HINDENNACH M., SCHENK A., HAHN H., PEITGEN H.-O.: Interactive determination of robust safety margins for oncologic liver surgery. *International Journal of Computer Assisted Radiology and Surgery 4* (2009), 469–474.

[IO92] ILG M., OGNIEWICZ R.: The application of voronoi skeletons to perceptual grouping in line images. In *Proceedings of the 11th International Conference on Pattern Recognition* (August 1992), vol. 3, pp. 382–385.

[JH11] JASSI P., HAMARNEH G.: Vascusynth: Vascular tree synthesis software. *Insight Journal January-June* (2011), 1–12.

[JU04] JACOB M., UNSER M.: Design of steerable filters for feature detection using canny-like criteria. *IEEE Transaction on Pattern Analysis and Machine Intelligence 26*, 8 (Aug 2004), 1007–1019.

[KBT*12] KRETSCHMER J., BECK T., TIETJEN C., PREIM B., STAMMINGER M.: Reliable Adaptive Modelling of Vascular Structures with Non-Circular Cross-Sections. *Computer Graphics Forum 31*, 3 (2012), 1055–1064.

[KEV*03] KRISSIAN K., ELLSMERE J., VOSBURGH K., KIKINIS R., WESTIN C.-F.: Multiscale segmentation of the aorta in 3d ultrasound images. In *25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS)* (2003), pp. 638–641.

[KGSD95] KOLLER T., GERIG G., SZEKELY G., DETTWILER D.: Multiscale detection of curvilinear structures in 2-d and 3-d image data. In *Proceedings of the Fifth International Conference on Computer Vision* (Jun 1995), pp. 864–869.

[KLDW12] KEIL M., LAURA C. O., DRECHSLER K., WESARG S.: Ct and ultrasound liver vessel registration based on combined segmentation of b-mode and color flow. In *Proceedings of the 3rd Eurographics Workshop on Visual Computing for Biology and Medicine (VCBM)* (2012). Accepted for publication.

[Kle06] KLETTE G.: Branch voxels and junctions in 3d skeletons. In *Proceedings of the 10th International Workshop on Combinatorial Image Analysis (IWCIA)* (2006), Springer, pp. 34–44.

[KMA*00] KRISSIAN K., MALANDAIN G., AYACHE N., VAILLANT R., TROUSSET Y.: Model-based detection of tubular structures in 3d images. *Comput. Vision Image Understanding 80*, 2 (2000), 130–171.

[KMV*98] KRISSIAN K., MALADAIN G., VAILLANT R., TROUSSET Y., AYACHE N.: Model-based multiscale detection of 3d vessels. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (1998), pp. 722–727.

[KNNS99] KARCH R., NEUMANN F., NEUMANN M., SCHREINER W.: A three-dimensional model for arterial tree representation, generated by constrained constructive optimization. *Computers in Biology and Medicine 29* (1999), 19–38.

[KQ04] KIRBAS C., QUEK F.: A review of vessel extraction techniques and algorithms. *ACM Computing Surveys (CSUR) 36* (June 2004), 81–121.

[KRBWC03] KRETOWSKI M., ROLLAND Y., BÉZY-WENDLING J., COATRIEUX J.-L.: Physiologically based modeling of 3-d vascular networks and ct scan angiography. *IEEE Transactions on Medical Imaging 22*, 2 (February 2003), 248–257.

[Kre09] KREITZ C.: *Segmentvisualisierung der Leber für die intraoperative Navigation*. Master's thesis, Technische Universität Darmstadt, 2009.

[KSNN98] KARCH R., SCHREINER W., NEUMANN F., NEUMANN M.: Three-dimensional growth and optimization of arterial tree models. In *Proceedings of the American Medical Informatics Association Symposium (AMIA)* (1998).

[KTA09] KAFTAN J. N., TEK H., AACH T.: A two-stage approach for fully automatic segmentation of venous vascular structures in liver CT images. In *Proceedings of SPIE - Medical Imaging 2009: Image Processing* (February 2009), vol. 7259, p. 725911.

[KTS99] KITAOKA H., TAKAKI R., SUKI B.: A three-dimensional model of the human airway tree. *Journal of Applied Physiology 87*, 6 (December 1999), 2207–2217.

[KVPL04] KONRAD-VERSE O., PREIM B., LITTMANN A.: Virtual resection with a deformable cutting plane. In *Proceedings of Simulation und Visualisierung* (2004), pp. 203–214.

[KWT88] KASS M., WITKIN A., TERZOPOULOS D.: Snakes: Active contour models. *International Journal of Computer Vision 1*, 4 (1988), 321–331.

[KYT12] KAUL V., YEZZI A., TSAI Y.: Detecting curves with unknown endpoints and arbitrary topology using minimal paths. *IEEE Transactions on Pattern Analysis and Machine Intelligence PP*, 99 (2012), 1.

[KZZ*07] KAWAJIRI S., ZHOU X., ZHANG X., HARA T., FUJITA H., YOKOYAMA R., KONDO H., KANEMATSU M., HOSHI H.: Automated segmentation of hepatic vessel trees in non-contrast x-ray ct images. In *Proceedings of SPIE - Medical Imaging 2007: Image Processing* (2007), vol. 6512, p. 65123A.

[LABFL09] LESAGE D., ANGELINI E. D., BLOCH I., FUNKA-LEA G.: A review of 3d vessel lumen segmentation techniques: models, features and extraction schemes. *Med Image Anal 13*, 6 (Dec 2009), 819–845.

[LC87] LORENSEN W. E., CLINE H. E.: Marching cubes: A high resolution 3d surface construction algorithm. *ACM SIGGRAPH Computer Graphics 21*, 4 (1987), 163–169.

[LC06] LAU K., CHUNG A.: A global optimization strategy for 3d-2d registration of vascular images. In *BMVC06* (September 2006), vol. 2, p. 489ff.

[LCB*97] LORENZ C., CARLSEN I.-C., BUZUG T. M., FASSNACHT C., WEESE J.: Multi-scale line segmentation with automatic estimation of width, contrast and tangential direction in 2d and 3d medical images. In *Proc. CVRMed-MRCAS* (1997), Springer, pp. 233–242.

[Lin99] LINDEBERG T.: Principles for automatic scale selection. In *Handbook on computer vision and applications* (1999), Jahne B., Haussecker H. W., Geissler P., (Eds.), vol. 2, Academic Press, Boston, USA, pp. 239–274.

[LJB08] LÄTHÉN G., JONASSON J., BORGA M.: Phase based level set segmentation of blood vessels. In *Proc ICPR* (Dec. 2008), pp. 1–4.

[LJB10] LÄTHÉN G., JONASSON J., BORGA M.: Blood vessel segmentation using multi-scale quadrature filtering. *Pattern Recognition Letters 31*, 8 (2010), 762–767. Award winning papers from the 19th International Conference on Pattern Recognition (ICPR), 19th International Conference in Pattern Recognition (ICPR).

[LKC94] LEE T.-C., KASHYAP R. L., CHU C.-N.: Building skeleton models via 3-d medial surface/axis thinning algorithms. *Graphical Models and Image Process (CVGIP) 56*, 6 (November 1994), 462–478.

[LKZ*08] LOHE T., KRÖGER T., ZIDOWITZ S., PEITGEN H.-O., JIANG X.: Hierarchical matching of anatomical trees for medical image registration. In *Proceedings of the First International Conference on Medical Biometrics (ICMB)* (2008), vol. 4901 of *Lecture Notes in Computer Science*, Springer, pp. 224–231.

[LLS92] LAM L., LEE S.-W., SUEN C. Y.: Thinning methodologies-a comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence 14* (September 1992), 869–885.

[LML10] LERMÉ N., MALGOUYRES F., LÉTOCART L.: Reducing graphs in graph cut segmentation. In *Proceedings of the International Conference on Image Processing (ICIP)* (2010).

[LnLByL07] LATECKI L., NAN LI Q., BAI X., YU LIU W.: Skeletonization using ssm of the distance transform. In *International Conference on Image Processing (ICIP)* (October 2007), vol. 5, pp. 349–352.

[LPH*09] LANGE T., PAPENBERG N., HELDMANN S., MODERSITZKI J., FISCHER B., LAMECKER H., SCHLAG P. M.: 3d ultrasound-ct registration of the liver using combined landmark-intensity information. *International Journal of Computer Assisted Radiology and Surgery (CARS) 4* (2009), 79–88.

[LRV*08] LEHMANN K. S., RITZ J.-P., VALDEIG S., SCHENK A., HOLMER C., PEITGEN H.-O., BUHR H.-J., FRERICKS B. B.: Portal vein segmentation of a 3d-planning system for liver surgery - in vivo evaluation in a porcine model. *Annals of Surgical Oncology 15*, 7 (2008), 1899–1907.

[LSGX05] LOMBAERT H., SUN Y., GRADY L., XU C.: A multilevel banded graph cuts method for fast image segmentation. In *Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV)* (Washington, DC, USA, 2005), IEEE Computer Society, pp. 259–265.

[LWRS09] LANGE T., WÖRZ S., ROHR K., SCHLAG P. M.: Landmark-based 3d elastic registration of pre- and postoperative liver ct data. In *BVM* (2009).

[LY07] LI H., YEZZI A.: A.: Vessels as 4-d curves: Global minimal 4-d paths to extract 3-d tubular surfaces and centerlines. *IEEE Transactions on Medical Imaging 26* (2007), 1213–1223.

[MBR*11] MOLTZ J., BRAUNEWELL S., RUHAAK J., HECKEL F., BARBIERI S., TAUTZ L., HAHN H., PEITGEN H.-O.: Analysis of variability in manual liver tumor delineation in ct scans. In *IEEE International Symposium on Biomedical Imaging: From Nano to Macro* (30 2011-april 2 2011), pp. 1974 –1977.

[Mei12] MEIXNER S.: *Konzeption und Realisierung einer Testumgebung zum standardisierten Evaluieren von Gefäßsegmentierungs-Algorithmen*. Bachelor's thesis, Hochschule Darmstadt, 2012.

[MHB09] MIRHASSANI S., HOSSEINI M., BEHRAD A.: Improvement of hessian based vessel segmentation using two stage threshold and morphological image recovering. In *International Conference on Innovations in Information Technology, 2009. IIT '09.* (15-17 2009), pp. 50 –54.

[MKS*07] METZEN J. H., KRÖGER T., SCHENK A., ZIDOWITZ S., PEITGEN H.-O., JIANG X.: Matching von baumstrukturen. In *BVM* (Munich, Germany, March 2007), Springer Verlag, pp. 116–120.

[ML09] MUJA M., LOWE D. G.: Fast approximate nearest neighbors with automatic algorithm configuration. In *International Conference on Computer Vision Theory and Application VISSAPP'09)* (2009), INSTICC Press, pp. 331–340.

[MM08] MUELLER D., MAEDER A.: Robust semi-automated path extraction for visualising stenosis of the coronary arteries. *Computerized Medical Imaging and Graphics 32*, 6 (2008), 463–475.

[MSS*04] MEINZER H.-P., SCHEMMER P., SCHÖBINGER M., NOLDEN M., HEIMANN T., YALCIN B., RICHTER G. M., KRAUS T., BÜCHLER M., THORN M.: Computer-based surgery planning for living liver donation. In *Proceedings of the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* (2004).

[MSvdG*07] METZ C., SCHAAP M., VAN DER GIESSEN A., VAN WALSUM T., NIESSEN W.: Semi-automatic coronary artery centerline extraction in computed tomography angiography data. In *Proceedings of the 4th IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI)* (2007), pp. 856–859.

[MVN06] MANNIESING R., VIERGEVER M. A., NIESSEN W. J.: Vessel enhancing diffusion: A scale space representation of vessel structures. *Med. Image Anal. 10*, 6 (2006), 815 – 825.

[NKLR10] NAM W. H., KANG D.-G., LEE D., RA J. B.: Robust registration of 3-d ultrasound and ct images of the liver for image-guided intervention. In *Proceedings of the 2010 IEEE international Conference on Biomedical imaging (ISBI)* (Piscataway, NJ, USA, 2010), IEEE Press, pp. 1201–1204.

[NKM10] NIMURA Y., KITASAKA T., MORI K.: Blood vessel segmentation using line-direction vector based on hessian analysis. In *Proceedings of SPIE - Medical Imaging 2010: Image Processing* (2010), vol. 7623, p. 76233Q.

[NP11] NÉMETH G., PALÁGYI K.: Topology preserving parallel thinning algorithms. *International Journal of Imaging Systems and Technology 21*, 1 (2011), 37–44.

[NSM05] NUMMINEN K., SIPILÄ O., MÄKISALO H.: Preoperative hepatic 3d models: Virtual liver resection using three-dimensional imaging technique. *European Journal of Radiology 56*, 2 (2005), 179–184.

[OBN03] OLABARRIAGA S., BREEUWER M., NIESSEN W.: Evaluation of hessian-based filters to enhance the axis of coronary arteries in ct images. In *Proceedings of the 17th International Congress and Exhibition Computer Assisted Radiology and Surgery* (2003), vol. 1256, pp. 1191–1196.

[Oel11] OELMANN S.: *Konzeption und Realisierung eines Verfahrens zur automatischen Erkennung von fehlerhaften Verbindungsstellen in einer Graphenrepräsentation des Gefäßsystems der Leber*. Bachelor's thesis, Hochschule Darmstadt, 2011.

[OFC11] OLIVEIRA D., FEITOSA R., CORREIA M.: Segmentation of liver, its vessels and lesions from ct images for surgical planning. *BioMedical Engineering OnLine 10* (2011), 30.

[OI92] OGNIEWICZ R., ILG M.: Voronoi skeletons: Theory and applications. In *Proceedings of the Conference on Computer Vision and Pattern Recognition* (1992), pp. 63–69.

[OLBPDW12] OYARZUN LAURA C., BUENO PLAZA P., DRECHSLER K., WESARG S.: Comparison between rigid registration methods in four clinical scenarios. In *Proceedings of the International MICCAI Workshop on Clinical Image-based Procedures: From Planning to Intervention* (2012), LNCS, Springer. Accepted for publication.

[OLD10] OYARZUN LAURA C., DRECHSLER K.: Efficient globally optimal matching of anatomical trees of the liver. In *Proceedings of the 2nd Eurographics Workshop on Visual Computing for Biology and Medicine (VCBM)* (2010), pp. 75–82.

[OLD11a] OYARZUN LAURA C., DRECHSLER K.: Computer assisted matching of anatomical vessel trees. *Computers & Graphics 35*, 2 (2011), 299–311.

[OLD11b] OYARZUN LAURA C., DRECHSLER K.: Extracting anatomical landmarks of the liver vasculature using tree matching. In *Biomedical Engineering* (2011), vol. 56.

[OLD11c] OYARZUN LAURA C., DRECHSLER K.: Graph to graph matching: Facing clinical challenges. In *Proceedings of the 24th IEEE International Symposium on Computer-Based Medical Systems (CBMS)* (2011).

[OLDE*12] OYARZUN LAURA C., DRECHSLER K., ERDT M., KEIL M., NOLL M., BENI S. D., SAKAS G., SOLBIATI L.: Intraoperative registration for liver tumor ablation. In *Abdominal Imaging. Computational and Clinical Applications* (2012), pp. 133–140.

[Ots79] OTSU N.: A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics 9*, 1 (January 1979), 62–66.

[Pal01] PALAGYI K.: A 3d parallel shrinking algorithm. *Acta Cybernetica 15* (2001), 2001.

[PBB05a] POCK T., BEICHEL C. J. R., BISCHOF H.: Multiscale medialness for robust segmentation of 3d tubular structures. In *Proceedings of the 10th Computer Vision Winter Workshop* (February 2005).

[PBB05b] POCK T., BEICHEL R., BISCHOF H.: A novel robust tube detection filter for 3d centerline extraction. In *Proc. SCIA* (2005).

[PD01] PAN S., DAWANT B. M.: Automatic 3d segmentation of the liver from abdominal ct images: a level-set approach. In *Proceedings of SPIE* (2001), pp. 128–138.

[PDB*09] PETERHANS M., DAGON B., BERG A. V., INDERBITZIN D., BAUR C., WEBER S.: A porcine liver model for validation of registration accuracy in image-guided surgery. In *BVM* (2009).

[PK98] PALAGYI K., KUBA A.: A hybrid thinning algorithm for 3d medical images. *Journal of Computing and Information Technology 6* (1998), 149–164.

[PK99a] PALAGYI K., KUBA A.: Directional 3d thinning using 8 subiterations. In *Proceedings of the International Conference on Discrete Geometry for Computer Imagery (DGCI)* (1999), Bertrand G., Couprie M., Perroton L., (Eds.), vol. 1568 of *Lecture Notes in Computer Science*, Springer Verlag Berlin Heidelberg, pp. 325–336.

[PK99b] PALAGYI K., KUBA A.: A parallel 3d 12-subiteration thinning algorithm. *Graphical Models and Image Processing 61* (1999), 199–221.

[PLS09] PINHO R., LUYCKX S., SIJBERS J.: Robust region growing based intrathoracic airway tree segmentation. In *Proceedings of the 2nd International Workshop on Pulmonary Image Analysis* (2009).

[PM90] PERONA P., MALIK J.: Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence 12*, 7 (July 1990), 629–639.

[Poc04] POCK T. G.: *Robust Segmentation of Tubular Structures in 3D Volume Data*. Master's thesis, Graz, University of Technology, December 2004.

[PRB*05] PASSAT N., RONSE C., BARUTHIO J., ARMSPACH J.-P., MAILLOT C., JAHN C.: Region-growing segmentation of brain vessels: an atlas-based automatic approach. *Journal of Magnetic Resonance Imaging 21*, 6 (Jun 2005), 715–725.

[PSB*01] PALAGYI K., SORANTIN E., BALOGH E., KUBA A., HALMAI C., ERDOHELYI B., HAUSEGGER K.: A sequential 3d thinning algorithm and its medical applications. In *Proceedings of the International Conference on Information Processing in Medical Imaging (IPMI)* (2001), Insana M., Leahy R., (Eds.),

vol. 2082 of *Lecture Notes in Computer Science*, Springer Verlag Berlin Heidelberg, pp. 409–415.

[PSV06] PINHO R., SIJBERS J., VOS W.: Efficient approaches to intrathoracic airway tree segmentations. *Belgian Day on Biomedical Engineering IEEE/EMBS Benelux Symposium 2* (2006).

[PTHS06] PALAGYI K., TSCHIRREN J., HOFFMAN E. A., SONKA M.: Quantitative analysis of pulmonary airway tree structure. *Computers in Biology and Medicine 36* (2006), 974–996.

[PTS03] PALAGYI K., TSCHIRREN J., SONKA M.: Quantitative analysis of three-dimensional tubular tree structures. In *Medical Imaging 2004: Image Processing, Proceedings of SPIE* (2003), vol. 5032, pp. 227–287.

[PVWL12] PAMULAPATI V., VENKATESAN A., WOOD B., LINGURARU M.: Liver segmental anatomy and analysis from vessel and tumor segmentation via optimized graph cuts. In *Abdominal Imaging. Computational and Clinical Applications*, Yoshida H., Sakas G., Linguraru M., (Eds.), vol. 7029 of *LNCS*. Springer, 2012, pp. 189–197.

[PWL11] PAMULAPATI V., WOOD B., LINGURARU M.: Intra-hepatic vessel segmentation and classification in multi-phase ct using optimized graph cuts. In *IEEE International Symposium on Biomedical Imaging: From Nano to Macro* (30 2011-april 2 2011), pp. 1982 –1985.

[QBD*09] QIAN X., BRENNAN M. P., DIONE D. P., DOBRUCKI W. L., JACKOWSKI M. P., BREUER C. K., SINUSAS A. J., PAPADEMETRIS X.: A non-parametric vessel detection method for complex vascular structures. *Med. Image Anal. 13*, 1 (2009), 49–61. Includes Special Section on Medical Image Analysis on the 2006 Workshop Microscopic Image Analysis with Applications in Biology.

[RBBS06] REITINGER B., BORNIK A., BEICHEL R., SCHMALSTIEG D.: Liver surgery planning using virtual reality. *IEEE Computer Graphics and Applications 26*, 6 (Nov.-Dec. 2006), 36–47.

[RBC*11] ROSSANT F., BADELLINO M., CHAVILLON A., BLOCH I., PAQUES M.: A morphological approach for vessel segmentation in eye fundus images, with quantitative evaluation. *Journal of Medical Imaging and Health Informatics 1*, 1 (2011), 42–49.

[RGP*06] RUTKAUSKAS S., GEDRIMAS V., PUNDZIUS J., BARAUSKAS G., BASEVICIUS A.: Clinical and anatomical basis for the classification of the structural parts of liver. *Medicina (Kaunas, Lithuania) 42*, 2 (2006), 98–106.

[SAS*07] SCHMIDT G., ATHELOGOU M., SCHÖNMEYER R., KORN R., BINNIG G.: Cognition network technology for a fully automated 3-d segmentation of liver. In *Proceedings of the MICCAI Workshop 3-D Segmentat. Clinic: A Grand Challenge* (2007), pp. 125–133.

[SCG*08] SHANG Q., CLEMENTS L., GALLOWAY R. L., CHAPMAN W. C., DAWANT B. M.: Adaptive directional region growing segmentation of the hepatic vasculature. In *Proceedings of SPIE Medical Imaging* (2008), vol. 6914, p. 69141F.

[Sch07] SCHULTE J.: *Ein Monitor-Service für Selbstorganisierende Systeme*. Bachelor's thesis, Technische Universität Kaiserslautern, 2007.

[SCW*11] SONG X., CHENG M., WANG B., HUANG S., HUANG X.: Computer-aided preoperative planning for liver surgery based on ct images. *Proceedings of the International Conference on Advances in Engineering 24*, 0 (2011), 133–137.

[SDB08] STAWIASKI J., DECENCIÉRE E., BIDAULT F.: Interactive liver tumor segmentation using graph-cuts and watershed. In *MICCAI Workshop on 3D Segmentation in the Clinic: A Grand Challenge II. Liver Tumor Segmentation Challenge* (2008).

[SDB*10] STALLKAMP J., DRECHSLER K., BERGEN T., KALTENBACHER D., BURISCH M., KAGE A., MÜNZENMAYER C., SAKAS G., WERNER N., WECHSLER A., WINTER C., WITTENBERG T.: Whole'o'hand - a holistic intervention and interaction system: A novel concept for closed-loop liver surgery. *Biomedizinische Technik / Biomedical Engineering 55*, Suppl. 1 (2010), 27–31.

[SDB*11] STALLKAMP J., DRECHSLER K., BERGEN T., KALTENBACHER D., BURISCH M., KAGE A., MÜNZENMAYER C., SAKAS G., WERNER N., WECHSLER A., WINTER C., WITTENBERG T.: Ein neues konzept eines integrierten assistenzsystems für die navigierte weichteilchirurgie. *Endoskopie heute 24*, 1 (2011), P10.

[SDM*01] SOLER L., DELINGETTE H., MALANDAIN G., MONTAGNAT J., AYACHE N., KOEHL C., DOURTHE O., MALASSAGNE B., SMITH M., MUTTER D., MARESCAUX J.: Fully automatic anatomical, pathological, and functional segmentation from ct scans for hepatic surgery. *Computer Aided Surgery 6*, 3 (2001), 131–42.

[Sel99] SELLE D.: *Analyse von Gefäßtrukturen in medizinischen Schichtdatensätzen für die computergestützte Operationsplanung*. PhD thesis, Universität Bremen, 1999.

[Set01] SETHIAN J.: Evolution, implementation, and application of level set and fast marching methods for advancing fronts. *Journal of Computational Physics 169*, 2 (2001), 503–555.

[SFM*10] SBOARINA A., FORONI R., MINICOZZI A., ANTIGA L., LUPIDI F., LONGHI M., GANAU M., NICOLATO A., RICCIARDI G., FENZI A., GEROSA M., DE SIMONE A., FRACASTORO G., GUGLIELMI A., CORDIANO C.: Software

for hepatic vessel classification: feasibility study for virtual surgery. *International Journal of Computer Assisted Radiology and Surgery 5* (2010), 39–48. 10.1007/s11548-009-0380-4.

[SGLL10] SALAZAR-GONZALEZ A., LI Y., LIU X.: Retinal blood vessel segmentation via graph cut. In *Proceedings of the 11th International Conference on Control Automation Robotics Vision (ICARCV)* (dec. 2010), pp. 225 –230.

[SH12] STEGER T., HOSSBACH M.: Navigated bronchoscopy using intraoperative fluoroscopy and preoperative CT. In *Proceedings of the 2012 IEEE international conference on Biomedical imaging: From nano to Macro* (2012), ISBI'12, IEEE Press.

[Sha10] SHANG Q.: *Separation And Segmentation of the Hepatic Vasculature In CT Images*. PhD thesis, Vanderbilt University, 2010.

[SHP11] SCHENK A., HAEMMERICH D., PREUSSER T.: Planning of image-guided interventions in the liver. *Pulse, IEEE 2*, 5 (sept.-oct. 2011), 48 –55.

[SMHS09] SHIKATA H., MCLENNAN G., HOFFMAN E. A., SONKA M.: Segmentation of pulmonary vascular trees from thoracic 3d ct images. *Int. J. Biomed. Imaging 2009* (2009), Article ID 636240, 11 pages.

[SMS*10] SCHWAIGER J., MARKERT M., SEIDL B., SHEVCHENKO N., DOERFLER N., LUETH T.: Risk analysis for intraoperative liver surgery. In *Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (31 2010-sept. 4 2010), pp. 410 –413.

[SNA*97] SATO Y., NAKAJIMA S., ATSUMI H., KOLLER T., GERIG G., YOSHIDA S., KIKINIS R.: 3d multi-scale line filter for segmentation and visualization of curvilinear structures in medical images. In *Proceedings of the First Joint Conference on Computer Vision, Virtual Reality and Robotics in Medicine and Medical Robotics and Computerassisted Surgery (CVRMed-MRCAS)* (1997), pp. 213–222.

[SNS*98] SATO Y., NAKAJIMA S., SHIRAGA N., ATSUMI H., YOSHIDA S., KOLLER T., GERIG G., KIKINIS R.: Three-dimensional multi-scale line filter for segmentation and visualization of curvilinear structures in medical images. *Med. Image Anal. 2*, 2 (June 1998), 143–168.

[SP09] SEO J.-J., PARK J.-W.: Automatic segmentation of hepatic vessels in abdominal mdct image. In *Proceedings of the Fourth International Conference on Computer Sciences and Convergence Information Technology* (Los Alamitos, CA, USA, 2009), IEEE Computer Society, pp. 420–424.

[SPSP02] SELLE D., PREIM B., SCHENK A., PEITGEN H.-O.: Analysis of vasculature for liver surgical planning. *IEEE Transactions on Medical Imaging 21*, 11 (2002), 1344–1357.

[SS04] SEZGIN M., SANKUR B.: Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging 13*, 1 (2004), 146–168.

[SS12] STEGER S., SAKAS G.: Fist: Fast interactive segmentation of tumors. In *Proceedings of Third International MICCAI Workshop on Abdominal Imaging: Computational and Clinical Applications* (2012).

[SSE*05] SEKIGUCHI H., SUGIMOTO N., EIHO S., HANAKAWA T., URAYAMA S.: Blood vessel segmentation for head mra using branch-based region growing. *Journal of Systems and Computers in Japan 36* (May 2005), 80–88.

[SSPP00] SELLE D., SPINDLER W., PREIM B., PEITGEN H.-O.: Mathematical methods in medical imaging: Analysis of vascular structures for liver surgery planning. In *Mathematics Unlimited - 2001 and Beyond* (2000), Springer, pp. 103–109.

[SSS*10] SHEVCHENKO N., SEIDL B., SCHWAIGER J., MARKERT M., LUETH T.: Mimed liver: A planning system for liver surgery. In *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE* (September 2010), pp. 1882 –1885.

[Str10] STROSCHE M.: *Interaktive Tumor-Segmentierung in medizinischen Volumendaten*. Bachelor's thesis, Technische Universität Darmstadt, 2010.

[SWB*08] SORANTIN E., WERKGARTNER G., BEICHEL R., BORNIK A., REITINGER B., POPOVIC N., SONKA M.: Virtual liver surgery planning. In *Image Processing in Radiology*, Medical Radiology. Springer, 2008, pp. 411–418.

[TBD11] TAUSCH R., BURISCH M., DRECHSLER K.: Registration of structured-light 3d scan data with ct skin information for patient positioning. In *Biomedical Engineering* (2011), vol. 56.

[tBSSS09] TEN BRINKE G., SLUMP C., STORM C., STOEL M.: A method towards automated thrombolysis in myocardial infarction (timi) frame counting using 3d reconstruction. In *Computers in Cardiology, 2009* (sept. 2009), pp. 653 –656.

[THMS05] TSCHIRREN J., HOFFMAN E. A., MCLENNAN G., SONKA M.: Intrathoracic airway trees: Segmentation and airway morphology analysis from low-dose ct scans. *IEEE Transactions on Medical Imaging 24*, 12 (December 2005), 1529–1539.

[TLG*05] TURGEON G.-A., LEHMANN G., GUIRAUDON G., DRANGOVA M., HOLDSWORTH D., PETERS T.: 2d-3d registration of coronary angiograms for cardiac procedure planning and guidance. *Medical Physics 32*, 12 (December 2005), 3737–3749.

[vLNHF95] VAN LEEUWEN M., NOORDZIJ J., HENNIPMAN A., FELDBERG M.: Planning of liver surgery using three dimensional imaging techniques. *European Journal of Cancer 31* (1995), 1212–1215.

[VS12] VIJAYAKUMARI V., SURIYANARAYANAN N.: Survey on the detection methods of blood vessel in retinal images. *European Journal of Scientific Research 68*, 1 (2012), 83–92.

[WDK*09] WITTENBERG T., DRECHSLER K., KALTENBACHER D., FRIEDL S., REIS C., SAKAS G., STALLKAMP J., KONDRUWEIT M.: 'miss heart': Assisting systems for minimal invasive smart suturing in cardiac surgery - a conceptually closed-loop approach. In *Proceedings of the 11th International World Congress on Medical Physics and Biomedical Engineering* (2009), Dössel O., Schlegel W., (Eds.), Springer, pp. 445–448.

[WF04] WESARG S., FIRLE E. A.: Segmentation of vessels: The corkscrew algorithm. In *Medical imaging 2004: Image processing.* (2004), vol. 5370, pp. 1609–1620.

[WKF06] WESARG S., KHAN M. F., FIRLE E.: Localizing calcifications in cardiac ct data sets using a new vessel segmentation approach. *Journal of Digital Imaging 19* (2006), 249–257.

[WKJL08] WESARG S., KHAN M., JAEHNE M., LACALLI C.: Automatisierte analyse der koronararterien basierend auf msct-daten. *Deutsche Zeitschrift fuer klinische Forschung 11*, 3/4 (2008), 28–33.

[XOR*00] XAVIER R. T., ORRIOLS X., RADEVA P., BINEFA X., VITRIÀ J., OERO C. C., VILLANUEVA J. J.: Eigensnakes for vessel segmentation in angiography. In *Proceedings of the International Conference on Pattern Recognition* (2000), IEEE, pp. 340–343.

[YBE11] YOU S., BAS E., ERDOGMUS D.: Extraction of samples from airway and vessel trees in 3d lung ct based on a multi-scale principal curve tracing algorithm. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (30 2011-sept. 3 2011), pp. 5157 –5160.

[YCHL12] YUAN F., CHI Y., HUANG S., LIU J.: Modeling n-furcated liver vessels from a 3-d segmented volume using hole-making and subdivision methods. *IEEE Transactions on Biomedical Engineering 59*, 2 (feb. 2012), 552–561.

[YMLB05] YU G., MIAO Y., LI P., BIAN Z.: Multiscale vessel segmentation: A level set approach. In *Progress in Pattern Recognition, Image Analysis and Applications* (2005), pp. 701–709.

[ZAGW07] ZÜHLKE D., ARNOLD S., GRUNST G., WISSKIRCHEN P.: Intra-interventional registration of 3d ultrasound to models of the vascular system of the liver. *GMS Computer- and Robot-Assisted Surgery (CURAC) 2*, 1 (December 2007), 1–6.

[ZCS*07] ZHOU C., CHAN H.-P., SAHINER B., HADJIISKI L. M., CHUGHTAI A., PATEL S., WEI J., GE J., CASCADE P. N., KAZEROONI E. A.: Automatic multiscale enhancement and segmentation of pulmonary vessels in ct pulmonary angiography images for cad applications. *Med. Phys. 34*, 12 (December 2007), 4567–4577.

[ZPFLSZ*11] ZHAN-PENG H., FA-LING Y., SHI-ZHONG J., ZHAO J., SU-SU B.: The segmentation of liver and vessels in ct images using 3d hierarchical seeded region growing. In *Proceedings of the IEEE International Conference on Computer Science and Automation Engineering (CSAE)* (june 2011), vol. 2, pp. 264 –269.